МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра загальної математики

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ КРИПТОГРАФІЇ ТА КРИПТОАНАЛІЗУ, ЩО ВИКОРИСТОВУЮТЬ АПАРАТ МОДУЛЯРНОЇ АРИФМЕТИКИ»**

Виконала: студентка 2 курсу, групи 8.1119
спеціальності 111 математика
(шифр і назва спеціальності)
освітньої програми математика
(назва освітньої програми)
В. І. Мерзлікіна
(ініціали та прізвище)

Керівник завідувач кафедри загальної математики, доцент, к.ф.-м.н. Зіновєєв І. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент доцент кафедри програмної інженерії, доцент, к.т.н., Мухін В. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя
2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра загальної математики

Рівень вищої освіти магістр

Спеціальність 111 математика

(шифр і назва)

Освітня програма математика

ЗАТВЕРДЖУЮ

Завідувач кафедри загальної математики, к.ф.-м.н., доцент

_____ І. В. Зіновєєв

(підпис)

«_____» _____2020 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Мерзлікіній Валерії Ігорівні

(Прізвище, ім'я та по-батькові)

1. Тема роботи Дослідження та практична реалізація алгоритмів криптографії та криптоаналізу, що використовують апарат модулярної арифметики

керівник роботи Зіновєєв Ігор Валерійович, к.ф.-м.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затвердженні наказом ЗНУ від «___» _____2020 р. №_____

2. Строк подання студентом роботи 11. 12. 2020 р.

3. Вихідні дані до роботи
   1. Постановка задачі
   2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Постановка задачі.
   2. Основні теоретичні відомості з криптографії та криптоаналізу.
   3. Огляд окремих алгоритмів криптографії.
   4. Огляд алгоритмів криптографії, що використовують апарат модулярної арифметики.

5. Перелік графічного матеріалу (з точним значенням обов'язкових креслень) Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 22. 05. 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	22.05.2020 – 28.05.2020	
2.	Збір вихідних даних.	29.05.2020 – 12.06.2020	
3.	Обробка методичних та	13.06.2020 – 25.06.2020	
	теоретичних джерел.		
4.	Розробка першого розділу.	26.06.2020 – 28.07.2020	
5.	Розробка другого розділу.	29.07.2020 – 01.09.2020	
6.	Розробка третього розділу.	02.09.2020 – 20.10.2020	
7.	Оформлення і нормоконтроль	21.10.2020 – 01.12.2020	
	кваліфікаційної роботи.		
8.	Захист кваліфікаційної роботи.	14.12.2020 - 18.12.2020	

Студент _____ В. І. Мерзлікіна
(підпис) (ініціали та прізвище)

Керівник роботи _____ І. В. Зіновєєв
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О. Г. Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Дослідження та практична реалізація алгоритмів криптографії та криптоаналізу, що використовують апарат модулярної арифметики»: 79 с., 10 рис., 13 табл., 34 джерел, 2 додатки.

ГАМУВАННЯ, ГЕНЕРАЦІЯ КЛЮЧІВ, ДЕШИФРУВАННЯ, КЛЮЧ, КРИПТОАНАЛІЗ, КРИПТОГРАФІЯ, КРИПТОСИСТЕМА, МОДУЛЯРНА АРИФМЕТИКА, НЕЗВЕДЕНІ МНОГОЧЛЕНИ, СКРЕМБЛЕР, ШИФР, ШИФРИ ГАМУВАННЯ.

Об’єкт дослідження – алгоритми криптографії та криптоаналізу.

Мета роботи: дослідити основні поняття криптографії, криптоаналізу та модулярної арифметики; дослідити класичні шифри криптографії, в алгоритмах яких використовується модулярна арифметика; дослідити сучасні алгоритми криптографії з використанням модулярної арифметики; навести та розв’язати приклади з застосуванням конкретного алгоритму з використанням модулярної арифметики; навести приклад практичної реалізації алгоритмів криптографії, що використовують апарат модулярної арифметики.

Методи дослідження – частково-пошуковий, дослідницький, аналітичний.

У кваліфікаційній роботі розглянуто основні поняття з криптографії, криптоаналізу та модулярної арифметики. Досліджено класичні шифри криптографії, серед яких виявлено шифри, алгоритми яких будуються на основі модулярної арифметики. Розглянуто сучасні алгоритми криптографії, що використовують апарат модулярної арифметики. Досліджено вид генерації ключів за допомогою незведених многочленів. Також ми дослідили принцип роботи скремблера та навели приклад практичної реалізації роботи скремблера.

SUMMARY

Master's Qualification Thesis «Research and the Implementation of the Cryptography and Cryptanalysis Algorithms, which Use the Modular Arithmetic Tool»: 79 pages,10 figures, 13 tables, 34 references, 2 supplements.

COLLECTION, GENERATION OF KEYS, DECYPRICATION, KEY, CRYPTOANALYSIS, CRYPTOGRAPHY, CRYPTOSYSTEM, MODULAR ARITHMETICS, NON-CONNECTED

The object of research is cryptographic algorithms.

Purpose: to explore the basic concepts of cryptography, cryptanalysis and modular arithmetic; to study classical cryptographic ciphers, in the algorithms of which modular arithmetic is used; to study modern cryptographic algorithms using modular arithmetic; give and solve examples using a specific algorithm using modular arithmetic; give an example of practical implementation of cryptography algorithms using the apparatus of modular arithmetic.

Research method – partially exploratory, research, analytical.

The basic concepts of cryptography, cryptanalysis and modular arithmetic are considered in the qualification work. Classical cryptographic ciphers have been studied, among which ciphers have been identified, the algorithms of which are based on modular arithmetic. Modern cryptography algorithms using modular arithmetic apparatus are considered. The type of key generation is studied, namely, the finding of new irreducible polynomials from a given undivided polynomial of the same degree under the condition that the roots of the polynomials are associated with arbitrary power dependences. We also researched the principle of scrambler operation and gave an example of practical implementation of scrambler operation.

ЗМІСТ

ВСТУП

Захист інформації перетворюється сьогодні на одну з найактуальніших задач внаслідок надзвичайно широкого розповсюдження як власне різноманітних систем обробки інформації, так і розширення локальних та глобальних комп'ютерних мереж, якими передаються величезні об'єми інформації державного, військового, комерційного, приватного характеру, власники якої часто були б категорично проти ознайомлення з нею сторонніх осіб. Надійні методи захисту від таких загроз надає криптографія.

Для професійного розуміння криптографічних алгоритмів і вміння оцінювати їх сильні і слабкі сторони, досліджувати і створювати нові шифри, необхідна математична підготовка. Для користувачів дуже важлива стійкість шифрів до зламу, і важливіше дізнатися про це раніше зловмисників. Створення надійних алгоритмів шифрування – складна задача, у розв'язку якої полягає якісно зіставлений алгоритм, в основу якого покладено математичну концепцію. Це пояснюється тим, що сучасна криптографія заснована на глибоких результатах таких розділів математики, як теорія складності обчислень, теорія чисел, теорія ймовірності, алгебра, теорія інформації та інші.

Серед алгоритмів шифрування важливу роль відіграють алгоритми, що побудовані на основі модулярної арифметики. Саме вони створюють складну криптосистему, на злам якої витрачається багато часу.

1 ОСНОВНІ ПОНЯТТЯ КРИПТОГРАФІЇ ТА МОДУЛЯРНОЇ АРИФМЕТИКИ

1.1 Означення та терміни криптографії

Мета нашого дослідження – дослідити використання модулярної арифметики в алгоритмах криптографії. Тому виникає необхідність дати визначення основним поняттям криптографії, для того щоб надалі користуватися ними. Будемо користуватися означеннями, які наведено у підручниках[1,2] та у криптографічних словниках [3].

Дамо визначення криптографії як науки:

Означення 1.1 Криптологія (математична криптографія) – галузь криптографії, математики і математичної кібернетики, яка вивчає математичні моделі криптографічних систем [1].

Означення 1.2 Криптографія – область наукових, прикладних, інженерно-технічних досліджень та практичної діяльності, яка пов’язана з розробкою методів криптографічного захисту інформації від загроз зі сторони противника або порушника, а також аналізом і обґрунтуванням їх криптографічної стійкості. Основними задачами криптографії є забезпечення конфіденційності, цілісності, аутентифікації [1].

Означення 1.3 Цілісність – відсутність змін у інформації, яку передають або зберігають у порівнянні з її вихідним записом. Необхідною умовою дотримання цілісності є захист повідомлення від навмисної або випадкової несанкціонованої модифікації або знищення [1].

Означення 1.4 Аутентифікація – встановлення (перевірка та підтвердження) справжності різних аспектів інформаційної взаємодії: змісту та джерела повідомлень, що передаються, сеансу зв’язку, часу взаємодії [1].

Означення 1.5 Конфіденційність – означає те, що інформацію призначено тільки певному колу осіб та яку потрібно зберігати у таємниці від інших [1].

Означення 1.6 Шифр – сімейство обернених відкритих відображень множини послідовностей блоків текстів (повідомлень) у множину послідовностей блоків текстів (повідомлень), що є зашифрованими, та які задаються за допомогою функцій шифрування. Математична модель шифру включає алгоритм зашифровування, алгоритм розшифровування, визначення режиму шифрування, а також модель множини відкритих текстів [2].

Означення 1.7 Шифротекст – текст, який отримано у результаті зашифровування відкритого тексту [2].

Означення 1.8 Зашифровування – процес перетворення відкритого повідомлення у шифроване повідомлення за допомогою ін’єктивної функції, що залежить від ключа з ключової множини (криптосистеми) [2].

Означення 1.9 Розшифровування – процес, зворотний до зашифровування, який реалізується за допомогою відомого параметру ключа.

Означення 1.10 Криптограма (шифрограма) – текст, який зашифровано [2].

Означення 1.11 Криптостійкість – характеристика шифру, яка визначає його стійкість до розшифровування. Найчастіше криптостійкість вимірюється кількістю операцій, які необхідні для перебору усіх можливих ключей, або інтервалом часу, необхідним для розшифровування [2].

Означення 1.12 Ключ – елемент (параметр), який змінюється і кожному значенню якого однозначно відповідає одне з відображень, що реалізується криптосистемою. Усі можливі значення ключа це ключова множина криптосистеми. Ключі можуть бути зіставними, тобто містити декілька частин, які забезпечують різні функції криптосистеми [3].

Означення 1.13 Відкритий ключ – несекретний ключ асиметричної шифросистеми [3].

Означення 1.14 Секретний ключ – ключ, який зберігається у таємниці від осіб, які не мають допуску до ключів даної симетричної шифросистеми або до використання деяких функцій даної шифросистеми [3].

Означення 1.15 Криптографічна система – система забезпечення безпеки інформації криптографічними методами у сенсі конфіденційності, цілісності, аутентифікації, неможливості відмови та невідстеження [3].

Існує декілька видів криптосистем, дамо визначення асиметричній та симетричній криптосистемам:

Означення 1.16 Асиметрична шифросистема – система шифрування, у якій асиметричним способом використовуються ключі двох видів – ключі відкриті та секретні [3].

Означення 1.17 Симетрична шифросистема – система шифрування, у якій симетричним способом використовуються секретні ключі зашифровування та ключі розшифровування. В такій системі ключі зашифровування та розшифровування у більшості випадків співпадають, а у окремих випадках один ключ легко визначається по іншому ключу [4].

Означення 1.18 Алгоритм шифрування –криптографічний алгоритм, який реалізує функцію шифрування [4].

Означення 1.19 Функція шифрування – функція, яка описує процес зашифровування та здійснює відкрите відображення послідовностей блоків тексту (повідомлень), яке залежить від ключа, у послідовності блоків тексту (повідомлень) шифрованого [4].

Означення 1.20 Злам шифру – процес отримання відкритого тексту з зашифрованого повідомлення без відомостей про шифр, який застосується [4].

Означення 1.21 Симетричні криптоалгоритми – алгоритми для шифрування і дешифрування яких використовується один і той же блок інформації (ключ) [4].

Означення 1.22 Асиметричні криптоалгоритми – алгоритми такі, що для шифрування повідомлення використовується один (відкритий) ключ,

відомий усім бажаючим, а для дешифрування – інший (закритий), який існує тільки в одержувача [5].

В залежності від характеру впливу на інформацію алгоритми підрозділяються на:

Означення 1.23 Перестановочні алгоритми – блоки інформації не змінюються самі по собі, але змінюється їх порядок проходження, що робить інформацію недоступною сторонньому спостерігачеві [5].

Означення 1.24 Підстановочні алгоритми – алгоритми, де самі блоки інформації змінюються за законами криптоалгоритмів. Переважна більшість сучасних алгоритмів належить цій групі [5].

Розглянемо основні поняття модулярної арифметики, а саме теорії чисел для подальшого застосування алгоритмів на основі модулярної арифметики в криптографії та криптоаналізі. Для цього звернемося до підручників [6,7], у яких наведені означення та формули з теорії чисел.

1.2 Означення та терміни модулярної арифметики

Означення 1.25 Модулярна арифметика – це система арифметики цілих чисел, в якій числа «обертаються навколо» деякого значення – модуля [6].

Модулярна арифметика пов'язана з залишком від ділення цілих чисел на певне задане натуральне число. Фактично в ній розглядаються класи еквівалентності певного натурального числа.

У сучасному вигляді модулярна арифметика була розвинута Карлом Фрідріхом Гаусом (1777-1855р.р.) – німецьким математиком.

Означення 1.26 Два цілих числа a, b називаються рівними за модулем n, якщо при діленні на ціле число на n вони мають однаковий залишок. Рівність чисел a і b за модулем n записують так [6]:

$$a \equiv b(mod\ n).$$

Еквівалентні означення [6]:

– різниця $a - b$ ділиться на n націло. Тобто, $a - b = kn$, де k – якесь ціле число;

– число a може бути записано у вигляді $a = b + kn$, де k – якесь ціле число;

Наприклад:

а) $15 \equiv 4(mod\ n)$. Справді, $15 - 4 = 11$ і 11 очевидно ділиться на 11;

б) $16 \equiv 37(\ mod\ n)$. Маємо $16 - 37 = -21$ і -21 ділиться на 7 націло.

Властивості, що виконуються для відношення рівності, виконуються також для рівності за модулем. Якщо $a_1 \equiv b_1\ (\ mod\ n)\ i\ a_2 \equiv b_2(mod\ n)$, тоді [6]:

$$(a_1 + a_2) \equiv (b_1 + b_2)(\ mod\ n),$$
$$(a_1 - a_2) \equiv (b_1 - b_2)(\ mod\ n),$$
$$(a_1 \cdot a_2) \equiv (b_1 \cdot b_2)(\ mod\ n).$$

З визначення рівності за модулем витікають такі властивості [6]:

а) рефлексивність: $a \equiv a(mod\ n)$;

б) симетричність: $a \equiv b(mod\ n) \leftrightarrow b \equiv a(mod\ n)$;

в) транзитивність: $a \equiv b(mod\ n)\ \ i\ \ b \equiv c(mod\ n) \rightarrow a \equiv c(mod\ n)$.

Тобто відношення рівності за модулем є відношенням еквівалентності на множині цілих чисел Z. Тоді Z розбивається на класи еквівалентності.

Клас еквівалентності відношення рівності за модулем n до якого належить число a позначається $\overline{a_n}$. Так як, $n \equiv 0(mod\ n)$, то додати n, теж саме, що і додати 0. Тому клас числа [6]:

$$\overline{a_n} = \{a, a \pm n, a \pm 2n, a \pm 3n, \dots\} = \{\dots, a - 2n, a - n, a + n, a + 2n, \dots\}.$$

Множина класів рівності за модулем n позначається Z/nZ і за означенням це:$Z/nZ = \{\overline{a_n}|a\epsilon Z\}$ [6].

Коли $n \neq 0$, Z/nZ має n елементів, і може бути записано [6]:

$$Z/nZ = \{\overline{0_n}, \overline{1_n}, \overline{2_n}, \dots, \overline{n-1_n}\}.$$

Для цих класів можна задати операції додавання, віднімання, множення. Таким чином Z/nZ є комутативним кільцем.

Деякий елемент $\overline{m_n}$, має обернений елемент тоді і лише тоді коли m і n є взаємно простими числами. Справді, якщо m і n є взаємно простими, то тоді існують $a, b\epsilon\ Z$ такі, що $an + bm = 1$. Звідси:

$$an + bm \equiv 1(mod\ n) \rightarrow bm \equiv 1(mod\ n).$$

Навпаки, якщо $bm \equiv 1\ (mod\ n)$ для деякого b, то $an + bm = 1$ для деякого a, враховуючи взаємну простоту m і n. Відповідно, якщо n просте число, то Z/nZ є полем [6].

Розглянемо розв'язування порівнянь першого степеню над полем Z/nZ. Рівняння записується у вигляді [6]:

$$a \cdot x \equiv b(mod\ n). \qquad (1.1)$$

Знайти розв'язок (1.1), тобто знайти всі значення x, які задовольняють даному рівнянню.

Розв'язок (1.1) можна подати за допомогою формули [6]:

$$x \equiv b \cdot a^{\varphi(n)-1}(mod\ n), \qquad (1.2)$$

якщо $\text{НСД}(a, n) = 1$, (найбільший спільний дільник чисел a та n) тобто взаємно прості числа [7].

Тут $\varphi(n)$ – це функція Ейлера, яка дорівнює кількості натуральних чисел, не більших за n і взаємно простих з ним. Якщо НСД $(a,n) \neq 1$, порівняння або має не єдиний розв'язок, або не має розв'язків. Як легко побачити, порівняння $2 \cdot x \equiv 3(mod\ 4)$ не має розвязків на множині натуральних чисел [7].

Інше порівняння $4 \cdot x \equiv 6(mod\ 22)$ має два розв'язки $x = 7, x = 18$.

Справедлива мала теорема Ферма, яка стверджує що якщо n – просте число та НСД$(a,n) = 1$, то $a^{n-1} \equiv 1(mod\ n)$ [7].

Узагальнення малої теореми Ферма, отримане Ейлером, стверджує що якщо НСД$(a,n) = 1$, то $a^{n-1} \equiv 1(mod\ n)$. З урахуванням наведених вище фактів отримаємо, що найбільше просте значення $a^{-1} \equiv a^{\varphi(n)-1}(mod\ n)$.

Алгоритм Евкліда знаходження найбільшого спільного дільника. Найбільше ціле число, що ділить одночасно цілі числа a і b, називається їх найбільшим дільником і позначається НСД(a,b), або просто (a,b). Якщо $(a,b) = 1$, то a і b називаються взаємно простими числами [7].

Алгоритм Евкліда знаходження найбільшого спільного дільника двох цілих чисел полягає у зведенні наступної послідовності операцій ділення з остачею [7]:

$$a = q \cdot b + r, 0 \leq r < b,$$
$$b = q_1 \cdot r + r_1, 0 \leq r_1 < r,$$
$$r = q_2 \cdot r_1 + r_2, 0 \leq r_2 < r_1,$$
$$r_1 = q_3 \cdot r_2 + r_3, 0 \leq r_3 < r_2,$$
$$\dots$$
$$r_k = q_{k+2} \cdot r_{k+1} + r_{k+2}, 0 \leq r_{k+2} < r_{k+1},$$
$$\dots$$
$$r_1 = q_3 \cdot r_2 + r_3, 0 \leq r_3 < r_2.$$

де a, b цілі числа;

q – неповне частка;

r – остача від ділення числа a на число n.

Коректне завершення алгоритму гарантується тим, що остачі від ділення створюють строго спадну послідовність натуральних чисел. З наведених рівностей: $(a,b) = (b.r) = (r,r_1) = \cdots = (r_{n-1},r_n) = r_n$ [7]. Тому найбільший дільник чисел a і b співпадає з r_n.

Як наслідок з алгоритму Евкліда [7], можна отримати ствердження, що найбільший дільник цілих чисел a і b можу бути представлен у виді лінійної комбінації цих чисел, тобто існують цілі числа u і v такі, що справедлива рівність $a \cdot u + b \cdot v = r_n$.

Розглянемо розв'язування порівнянь другого степеню над полем Z/nZ. З порівнянь степеню $n > 1$ будемо розглядати тільки найпростіші, а саме – двочлені порівняння [7]:

$$x^n \equiv a(mod\ m), \quad (a,m) = 1. \tag{1.3}$$

Якщо порівняння (1.3) має розв'язок, тоді a називається лишком степеню n за модулем m. У протилежному випадку a називається нелишком степеню n за модулем m. Зокрема, при $n = 2$ лишки або нелишки називаються квадратичними, при $n = 3$ – кубічними , при $n = 4$ – біквадратичними [7].

Розглянемо конкретний випадок, коли $n = 2$ і в першу чергу розглянемо двочлені порівняння другого степеню за простим непарним модулем p [7]:

$$x^2 \equiv a(mod\ p),\ (a,p) = 1. \tag{1.4}$$

Якщо a – квадратичний лишок за модулем p, то порівняння (1.4) має два розв'язків.

Дійсно, якщо a – квадратичний лишок, то порівняння (1.4) має, принаймні, один розв'язок $x \equiv x_1(mod\ p)$. Але тоді, з огляду на $(-x_1^2) = x_1^2$, теж саме порівняння має і другий розв'язок $x \equiv -x_1(mod\ p)$. Цей другий

розв'язок відрізняється від першого, так як з $x_1 \equiv -x_1(\,mod\,p)$, ми мали б $2 \cdot x_1 \equiv 0(mod\,p)$, що неможливе, з огляду на $(2,p) = (x_1,p) = 1$ [7].

Вказаними двома розв'язками вичерпуються всі розв'язки (1.4), так як останнє, будучи порівнянням другого степеню, більш ніж два розв'язки мати не може.

Приведена система лишків за модулем p складається з $\frac{p-1}{2}$ квадратичних лишків, які порівнюються з числами

$$1^2, 2^2, \dots, \left(\frac{p-1}{2}\right)^2, \tag{1.5}$$

і $\frac{(p-1)}{2}$ квадратичних лишків [7].

Дійсно, серед лишків наведеної системи за модулем p квадратичними лишками є ті, і тільки ті, які можна порівняти з квадратами чисел.

$$-\frac{p-1}{2}, \dots, -2, -1, 1, 2, \dots, \frac{p-1}{2}, \tag{1.6}$$

тобто з числами (1.5).

При цьому числа (1.5) за модулем p не можна порівняти, так як з $k^2 \equiv l^2(mod\,p)$, $0 < k < l \leq \frac{p-1}{2}$, випливало б, що порівнянню $x^2 \equiv l^2(mod\,p)$, всупереч $x = -l, -k, k, l$.

Перейдемо до лінійних систем порівнянь. Для знаходження розв'язку такої системи звернемося до китайської теореми про залишки – один з основних результатів елементарної теорії чисел, отриманий у I ст. китайським математиком Сун Це. Використовуючи позначення модулярної арифметики її можна сформулювати наступним чином.

Нехай $m_1, m_2, \dots, m_t$ попарно взаємно прості числа. Тоді для ∀ цілих чисел $a_1, a_2, \dots, a_t$ порівняння [7]:

$$\begin{cases} \mathrm{x} \equiv a_1 (\bmod \mathrm{m}_1), \\ \mathrm{x} \equiv a_2 (\bmod \mathrm{m}_2), \\ \mathrm{x} \equiv a_3 (\bmod \mathrm{m}_3), \\ \quad \dots, \\ \mathrm{x} \equiv a_\mathrm{t} (\bmod \mathrm{m}_\mathrm{t}), \end{cases}$$

мають в інтервалі $[0, M-1], M = m_1 \cdot m_2 \cdot \dots \cdot m_t$ єдиний спільний розв'язок виду [7]:

$$x = \sum_{j=1}^{t} a_j \cdot N_j \cdot M_j (mod\ M),$$

де $M_j = \frac{M}{m_j}$, а $N_j = M_j^{-1} mod\ m_j, j = 1, \dots, t.$

Китайська теорема про залишки стверджує [7], що кільце лишків за модулем добутку кількох попарно взаємно простих чисел є прямим (декартовим) добутком відповідних множникам лишків.

1.3 Висновки за розділом 1

У розділі 1 ми дослідили основні поняття криптографії та криптоаналізу, основні означення криптографії та криптоаналізу. Розглянули основні поняття модулярної арифметики, а саме теорії чисел, які будуть необхідними для подальшого дослідження алгоритмів, які побудовано на основі модулярної арифметики.

2 ІСТОРИЧНИЙ ОГЛЯД РОЗВИТКУ КРИПТОГРАФІЇ ТА ЇЇ СУЧАСНИЙ СТАН

2.1 Історія виникнення і розвитку криптографії. Класичні шифри

Будемо досліджувати історію криптографії звернувшись до [8, 9, 10].

Криптографія почала розвиватися 4 тис. років назад. Вже в історичних документах древньої цивілізації – Індії, Єгипті, Китаї, Месопотамії – є довідки про системи та способи створення шифрувального листа. Періоди становлення криптографії можна поділити на чотири основних. І період (3 тис. р. до н. е.) – моноалфавітні шифри, тобто шифри, де одна літера алфавіту замінюється на іншу. ІІ період (з ІХ ст. на Ближньому Сході та з ХV ст. у Європі до початку ХХ ст.) – період поліалфавітних шифрів, тобто шифрів простої заміни, де застосовується цикл моноалфавітних шифрів до зазначеної кількості літер зашифрованого тексту. ІІІ період (початок ХХ ст. до середини ХХ ст.) – період створення електротехнічних засобів до шифрувального апарату. ІV період (з середини ХХ ст. до 70-х років ХХ ст.) – період застосування математичної концепції до криптографії, тобто з'являються математичні означення кількості інформації, передачі даних, ентропії, функції шифрування і тому подібне.

За ці періоди криптографія зазнала великих змін, за допомогою яких можна виділити способи задання шифрів: шифр заміни(підстановки), перестановки, аналітичне перетворення, гамування і комбіновані шифри.

Великі внески в історію криптографії зробили багато відомих особистостей. Розглянемо декілька класичних шифрів кожного періоду криптографії. Наприклад, перші відомості про використання шифру у воєнній справі пов'язані з ім'ям спартанського полководця Лисандра (шифр «сцитала») [8]. Це був жезл циліндричної форми, який обгортала стрічка з пергаменту. Вздовж осі циліндра на пергаменті записувався текст, який

зашифровувався. Такий шифр здійснював перестановку літер повідомлення. Ключом цього шифру був діаметр сцитали.

Грецький письменник Полібій використовував систему сигналізації [8], яка була розповсюджена як метод шифрування. Він записував літери алфавіту у квадратну таблицю і замінював їх координатами: парами чисел (i, j), де i – номер рядку, а j – номер стовпця. У основі цього шифру полягав латинський алфавіт [8].

Таблиця 2.1 – Квадрат Полібія

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пари (i, j) передавалися за допомогою факелів. Наприклад, для передачі літери О треба було взяти 3 факела у праву руку і 4 факела у ліву [7].

Подібні шифрувальні методи з деякими змінами проіснували до епохи військових походів Юлія Цезаря, де шифри створювалися за допомогою модулярної арифметики, що робило їх більш стійкими до атаки ворога. Застосування модулярної арифметики до алгоритмів криптографії має окремий розділ в криптографії [8].

Шифр Цезаря – симетричний алгоритм шифрування підстановками. Використовувався римським імператором Юлієм Цезарем для приватного листування [8].

Юлій Цезар використовував адитивний шифр, щоб зв’язатися зі своїми чиновниками. Із цієї причини адитивні шифри згадуються іноді як шифри Цезаря. Цезар для свого зв’язку використовував цифру 3 [8].

Принцип дії полягає в тому, щоб циклічно зсунити алфавіт, а ключ k— це кількість літер, на які робиться зсув. Мовою теорії чисел [8], а саме теорії порівнянь ця умова запишеться у вигляді

$$y = (k + x)(mod\ n),$$
$$x = (y - k)(mod\ n), \quad (1.7)$$

де x – це порядковий номер символу відкритого тексту, y – це порядковий номер символу шифрованого тексту, n – це потужність алфавіту, а k – ключ.

Можна помітити, що суперпозиція двох шифрувань на ключах $k_1\ i\ k_2$ є просто шифруванням на ключі $k_1 + k_2$. Більш загально, множина перетворень шифрування шифру Цезаря утворює групу Z_n.

Припустимо, що, використовуючи шифр Цезаря, з ключем, який дорівнює 4, необхідно зашифрувати слово «КРИПТОГРАФІЯ».

Для цього зрушимо алфавіт так, щоб він починався з п’ятої літери (Д). Отже, отримаємо:

АБВГДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ – вихідний алфавіт,

зміщуємо всі літери вліво на 4, відповідно отримаємо:

А Б В Г Д Е Є Ж З И І Ї Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я
Д Е Є Ж З И І Ї Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я А Б В Г,

де $Д = А, Е = Б, Є = В,$ і т. д.

Використовуючи цю схему, відкритий текст «КРИПТОГРАФІЯ» перетворюється на «ОФКУЦТЖФДШЛГ». Для того, щоб одержувач повідомлення зміг відновити вихідний текст, необхідно повідомити йому, що ключ $k = 4$.

Шифр Цезаря має замало ключів – на одиницю менше, ніж літер в абетці. Тому перебрати усі ключі не складає особливої складності. Дешифрування з одним з ключів дає нам вірний відкритий текст. Даний шифр є простим, отже його легко зламати, тому виникала необхідність надалі удосконалювати алгоритми шифрування [8].

З часів Цезаря до XV ст. криптографія змінилась, але на жаль, дуже мало відомостей про методи шифрування у цей період. Але як відомо, за допомогою застосування саме модулярної арифметики до шифрування тексту криптографія вже мала серйозні наміри щодо захисту інформації. Дешифрування такого шифру, що будувався на основі алгоритмів модулярної арифметики, вимагав багато часу і труднощів, а отже мав стійкість до зламу і надійність. У XIV ст. з'явилась книга про системи тайнопису, автором якої є Папа Римський Чікко Сімонетті. У цій книзі містяться приклади шифру заміни, у яких голосні літери відповідають декільком значковим виразам. Такі шифри пізніше отримали назву шифри многозначної заміни або омофони [9].

Ще один значний крок уперед криптографія зробила завдяки Леону Альберті. Відомий філософ, архітектор у 1466 р. написав працю про шифри, де було запропоновано шифр, в основу якого полягав шифрувальний диск. Також ідея шифрувального диску простежувалась у працях Енея Тактіка. Ім'я цього винахідника пов'язують з декількома техніками шифрування і тайнопису –диск Енея, лінійка Енея і книжковий шифр. Диск мав отвори для ниточки під кожною літерою, для запису тексту, нитка простягалася через потрібний отвір під літерою. Одержувач, що витягував нитку, міг прочитати текст, але в зворотному порядку, що було легко зробити. Лінійка Енея носила більш складний характер шифру, який реалізовував шифр заміни. Суть з ниткою і отворами така ж як і з диском, але одержувач повинен був мати точно таку ж лінійку з такими ж отворами, щоб прочитати текст. Що стосується книжкового шифру, то тут Еней запропонував робити малопомітні дірки поруч з літерою у книжці або іншому документі. Таку техніку тайнопису використовували навіть німецькі шпигуни під час Другої Світової війни [9].

Одне з найважливіших удосконалень багатоалфавітних систем, яка складається з ідеї використовування в якості ключа тексту самого повідомлення або ж шифрованого тексту, належить Джероламо Кардано і Безу де Віженеру. Такий шифр отримав назву самоключ. Цей шифр будувався за допомогою таблиці, і пізніше шифр отримав назву таблиця Віженера. В загальному випадку ця таблиця складалася з алфавітів, що циклічно зрушувалися, причому перший рядок може бути довільним змішаним алфавітом. Перший рядок слугує алфавітом відкритого тексту, а перший стовпець – алфавітом ключа. Для шифрування відкритого повідомлення ($P_0 = p_1 p_2 \ldots$) Віженер запропонував в якості ключової послідовності (H) використовувати саме повідомлення (P_0) з доданої до нього у якості першої літери (p_0), відомої відправнику і одержувачу [10]. Послідовності літер підписувалися один під одним:

$$H = p_0 p_1 p_2 \ldots p_{i-1} \ldots,$$
$$P_0 = p_1 p_2 p_3 \ldots p_i \ldots,$$
$$P_{\text{ш}} = m_1 m_2 m_3 \ldots m_i \ldots .$$

При цьому пара літер, що знаходяться одна під одною в H і P_0, вказує, відповідно, номери рядків і стовпців таблиці, на перетині якимх знаходиться знак m_i шифрованого тексту ($P_{\text{ш}}$). Приклад таблиці Віженера можна подивитись у додатку А [10].

На початку XVII ст. над розвитком криптографії працював Матео Ардженетті. Він уперше запропонував використовувати деяке слово у якості мнемонічного ключа для змішаного алфавіту.

В цілому можна сказати, що XVII та XVIII ст. не надали нових ідей для криптографії, але вже у XIX ст. ситуація змінилась на протилежне.

У 1917 році американський інженер Гілберт Вернам (1890-1960 р.р.) створив шифр, який мав неабияку стійкість до зламу. Це був поточний шифр над двійковим алфавітом з літерами 0 і 1. Принцип дії полягає у наступному:

відкритий текст подається у двійковому виді, ключ (k) якого є випадкова двійкова послідовність тієї ж довжини, що використовується тільки один раз – для шифрування даного тексту. У результаті отримана криптограма є додаванням по символам відкритого тексту та ключа за модулем 2. Зауважимо, що оскільки за модулем 2 віднімання співпадає з додаванням, для дешифрування криптограма додається по символам з ключом. Як приклад, візьмемо відкритий текст $m\ o\ d\ u\ l\ a\ r$. За допомогою телеграфної кодової таблиці Бодо.

Знаходимо відповідні позначення літер [10]: m – 11100, o – 11000, d – 01001, u – 00111, l – 10010, a – 00011, r – 01010, так що отримаємо шифрувальну двійкову послідовність довжиною 35: 11100110000100100111100100001101010. У якості ключа візьмемо двійковий запис цифр після коми у числа $\pi = 3{,}1415926536\ldots$. Для двійкового представлення будь-якого числа від 0 до 15 достатньо чотирьох цифр: 0 – 0000, 1 – 0001, 2 – 0010, 3 – 0011, 4 – 0100, 5 – 0101, 6 – 0110, 7 – 0111, 8 – 1000, 9 – 1001, …, 15 – 1111. Обираючи перші 35 двійкових знаків, які кодують послідовність 141592653, знаходимо ключ k:

$$k = 00010100000101011001001001100101001.$$

Для отримання криптограми по символам додаємо за модулем 2 двійкові коди відкритого тексту і ключа:

$$\begin{array}{r} 11100110000100100111100100001101010 \\ +_2 \\ 00010100000101011001001001100101001 \\ \hline = 11110010000001111110101101101000011 \end{array}$$

Зауважимо що при додаванні криптограми і ключа ми отримаємо відкритий текст.

Винахід у середині XIX ст. телеграфу та інших технічних видів зв'язку дало новий поштовх розвитку криптографії. Інформація представлялась у вигляді двійкового коду, який ми розглядали вище. Тому виникла проблема «раціонального» представлення інформації, яка вирішувалась за допомогою кодів. Коди дозволяли передавати довге слово або цілу фразу двома-трьома знаками. З'явилася необхідність у високошвидкісних засобах шифрування та у корекційних кодах, для зв'язку з неминучими помилками при передачі повідомлення. На цьому етапі розвитку відомий дисковий шифратор Т. Джефферсона [10], чия криптосистема мала велику кількість ключових елементів, та отримала назву блочні шифри. Чарльз Уітстон винайшов шифр, який отримав назву шифр Плейфера. Це був перший з відомих бігамних літерних шифрів. У другій половині XIX ст. з'явився вельми стійкий спосіб ускладнення числових кодів – гамування, метод симетричного шифрування. Він полягав у «накладанні» послідовності, яка насамперед складалася з випадкових чисел, на відкритий текст. Така послідовність отримала назву гамма-послідовність і швидко розповсюджувалась у використанні до шифрування і дешифрування текстів. Клод Шеннон, американський криптоаналітик, математик і інженер, довів, що такий метод шифрування є досить стійким до зламу [10].

2.2 Аналіз сучасного стану досліджень у криптографії та криптоаналізу

Теорія інформації у сучасному світі отримала початок розвитку з роботи Огюста Кергоффса «Військова криптографія», яка була опублікована ще у 1883 році. Після цієї роботи саме Клод Шеннон з роботою «Теорія зв'язку у секретних системах», 1949 року, висловив свою думку про необхідні та достатні умови шифрування і дешифрування. Ще довгий час криптографія була таємним джерелом інформації. Але з появою сучасних комп'ютерів

передача засекреченої інформації потребувала більш складних та стійких алгоритмів шифрування. Саме таким прикладом було використання симетричної та асиметричної криптографії.

При використанні алгоритмів шифрування виникає необхідність перевірки їх на стійкість до різноманітних крипто-атак. Одним з блочних алгоритмів шифрування є DES (Data Encryption Standard) – стандарт шифрування. Саме для аналізу цього алгоритму були створені лінійні атаки та диференціальний криптоаналіз, які використовували до цілого класу блочних шифрів.

На сьогоднішній день сучасні алгоритми блочного шифрування розробляються з розрахунком на те, що зловмисник має менше шансів знайти засекречений ключ, навіть якщо йому відомий алгоритм шифрування.

Сучасна криптографія заснована на понятті односпрямованої функції. Стійкість таких сучасних шифрів визначається довжиною використаного ключа шифрування.

Саме тому ми дослідили основні характерні особливості криптографічних систем, основні проблеми, які пов’язані з визначенням криптографічної стійкості сучасних криптосистем, та провели їх аналіз на основі сучасних статей.

При дослідженні аналізу криптографічних систем ми виявили що першим кроком є визначення набору даних, який відомий на початку. Якщо відомим є алгоритм шифрування і є хоча б одна пара відкритий – зашифрований текст, то використовується спосіб послідовного випробовування всіх можливих варіантів ключа. Таке випробовування триває поки інформація не буде дешифрована. Такий спосіб має декілька назв у літературі: «Метод повного перебору», «Метод грубої сили» або «Метод атаки в лоб». Головною перевагою цього методу є мінімальний набір даних, який необхідно для пошуку ключа. Але не завжди такий метод відповідає практичним вимогам.

На початку 90-х років метод лінійного криптоаналізу запропонував японський вчений М. Матсуі. Цей метод дозволяв проводити аналіз шляхом опробування 2^{47} пар текстів, які зашифровані на єдиному ключі. Порівняно з попереднім методом кількість опробування зменшилася, але виникло практично нездійснене – наявність великого об'єму інформації, яка зашифрована на єдиному ключі.

Після цього запропоновано метод диференціального аналізу Е. Біхамом та А. Шаміром. За допомогою цього методу складність аналізу скоротилася ще більше до 37. Надалі розвиток цих методів демонструє можливість їх застосування до цілого класу блочних шифрів, дозволяє виявити слабкі місця інших алгоритмів шифрування.

На сьогоднішній день існує вдосконалений лінійно-диференціальний метод, метод неможливих диференціалів, які використано для оцінювання стійкості інших шифрів. Про це йдеться у статті [11].

Симетрична криптографія. Для симетричних алгоритмів шифрування характерні наступні властивості: використання одного і того ж самого алгоритму як для шифрування, та і для дешифрування інформації; використання одного секретного ключа. Такі алгоритми поділяються на блочні та поточні. Для блочних алгоритмів шифрування інформації здійснюється блоками. Поточні шифри здійснюються в режимі реального часу, як правило, побітово та використовують для шифрування випадкову послідовність.

Розглянемо основні сучасні методи аналізу симетричних систем.

Диференціальний криптоаналіз. Метод диференціального аналізу, лінійно-диференціальний метод, метод неможливих диференціалів, метод бумеранга – використовуються для оцінки стійкості інших алгоритмів шифрування. Метод диференціального аналізу базується на аналізі блочного шифрування. Тобто для шифрування використано не один текст, а пара текстів, що ускладнює аналіз. Для тексту відмінності будуть лише у деяких позиціях. Тому, для визначення цієї різниці необхідно пару текстів скласти між собою за модулем 2. Як результат такого додавання ми отримаємо на

виході значення нуль у тих позиціях, в яких початкові тексти були однакові, та відповідно значення один у тих місцях, у яких початкові тексти мали різницю.

У загальному випадку диференціальний аналіз блочних алгоритмів шифрування має декілька етапів. Перший етап – теоретична задача, одноразове знаходження для алгоритму шифрування характеристик, які мають максимальні значення. Другий етап – обчислювальна стійка задача, пошук правильних пар текстів з використанням тих самих характеристик. І третій етап – послідовний алгоритм, аналіз правильних пар текстів та накопичення статистики про можливі значення секретного ключа шифрування.

У наступних статтях досліджено можливості застосування методу диференціального крипто аналізу до аналізу поточних шифрів та сучасних функцій хешування.

У статті [11] автор досліджує проведення атаки на блочні алгоритми шифрування з використанням методу диференціального криптоаналізу. В статті представлені основні означення, які широко використовуються при описі методу, який описано вище. Автор провів експеримент проведення атаки для шифру, який скорочено до трьох раундів.

Алгебраїчний аналіз. Сутність алгебраїчних методів аналізу заключено у отриманні рівнянь, які описують нелінійні перетворення заміни S –блоків, з наступними розв'язками знайдених систем рівнянь та отриманням ключа шифрування. Такий метод криптоаналізу здійснюється на основі атак з відомим відкритим текстом, для успішного аналізу достатньо мати одну пару відкритого тексту. Також як і диференціальний аналіз, алгебраїчний можна поділити на декілька етапів: перший етап – створення системи рівнянь, яка описує перетворення у нелінійних криптографічних примітивах шифру, який аналізується.

Другий етап – розв'язок отриманої системи рівнянь. Такий аналіз представлено у наступних статтях.

У статті [12] описано порівняльний аналіз алгоритмів для операцій зведення в степінь за модулем цілих чисел великої розрядності. Автор пропонує алгоритм, де використано набір передобчислювальних значень за фіксованою основою. Також у статті представлено паралельну модифікацію такого алгоритму. Приводяться результати застосування даних алгоритмів для найбільш відомих схем і алгоритмів криптографії.

У статті [13] описано метод криптографії, який дає змогу отримати нові незведені многочлени з даного незведеного многочлену того ж степеню, за умовою що корені цих многочленів пов’язані степеневою залежністю. Автор пропонує формули для генерації таких многочленів, які використовують апарат модулярної арифметики. Виявлено що такий метод широко застосовується у криптографічних додатках, для алгоритмізації побудови незведеного многочлена з загальним степеневим зв’язком коренів.

У статті [13] автор досліджує характеристичні многочлени деяких класів гіпереліптичних кривих над кінцевим полем. У статті отримано порівняння за модулем характеристики та обмеження на коефіцієнти для характеристичних многочленів кривих з автоморфізмами.

У статті гіпереліптичні криві представлені як альтернатива еліптичним кривим, що потребують менший розмір ключа при високому рівні безпеки. На основі представлених результатів у статті можна побудувати алгоритм підрахунку числа точок на кривих, що ізоморфні кривим з автоморфізмами над розширенням кінцевого поля, з метою використання у криптосистемах з високою криптостійкістю.

У статті [14] представлено нову семантично стійку систему шифрування з відкритим ключем на базі RSA. Автор досліджує переваги цієї криптосистеми такі як: стійкість, широкий вибір ключів та вибір ключа користувачем. Автор порівнює дану криптосистему з базовою системою RSA та виявляє її недоліки – недостатні знання про розкладання модуля на множники. Автор досліджує вибір та завдання складових даної

криптосистеми, властивості побудованої системи шифрування, шифрування на підгрупі квадратичних залишків.

З останнього автор розглядає використання квадратичної підсистеми RSA та виявляє ряд відмінностей: можливість використовувати парні ключі шифрування; дешифрування при використанні ключів виду 2^s і т. д.

В статті [15] представлено алгоритм шифрування даних з симетричним ключем з використанням комбінації практично необернених перетворень. Такий алгоритм засновано на мережі Фейстеля, який достатньо легко реалізується. Автор докладно розповідає в статті про математичну модель даного перетворення, його реалізацію, постановку задачі з комбінацією перетворень: бітовне додавання за модулем; матричне перетворення за модулем; шифрування 64-бітових блоків даних за допомогою 256-бітового ключа. Автор отримує такі результати: додавання відкритого тексту з раундовим ключем за модулем 2, логічне перетворення, матричне перетворення за модулем 256, S-блок та таблиця стиснення є практично необернеными. Перетворення за таблицею стиснення нелінійне, тобто є властивість неоднозначності у процесах шифрування та дешифрування. Нелінійність даної операції створена на основі повторення значення елементів з рівно випадковим розподіленням у конструкції таблиці. Перетворення гамування бітовим додаванням відкритого тексту з раундовим ключем за модулем 2 лінійне, але воно теж є практично необернеником через невідому інформацію, яка гамується. Автор наголошує що всі перераховані властивості перетворень функції мережі Фейстеля алгоритму шифрування даних у статті [16] забезпечать стійкість разом з невідомим ключем довжиною 256 біт.

Автор статті [16] досліджує основні особливості функціювання та реалізації поточного шифру TRIVIUM. Такий шифр є складовою малоресурсної криптографії. В статті представлено дослідження особливостей побудови алгоритму TRIVIUM, отримані характеристики програмної реалізації та основних блоків апаратної реалізації, з перспективами подальшого ефективного використовування. Автор пропонує програмну

модель шифру TRIVIUM, яка може бути використана для навчання студентів при викладанні сучасних поточних шифрів малоресурсної криптографії. Така програма призначена не тільки для створення зашифрованих документів відкритих текстів, але і дозволяє прослідкувати бітові взаємодії декількох тактів роботи програми. В статті розглянута можливість та розроблені основні блоки апартної реалізації алгоритму $TRIVIUM$ у довкіллі проектування $Qurtus\ II$.

Асиметрична криптографія. Алгоритми, які базуються на асиметричній криптографії, є більш стійкішими та практичними. Такі алгоритми будуються на розв’язку однієї зі складних математичних задач, таких як дискретне логарифмування або задача про факторизацію чисел. Для таких алгоритмів характерні наступні властивості: не обов’язково використання одного і того ж самого алгоритму як для шифрування, так і для дешифрування даних; використання двох ключей, один з яких є відкритим, а інший – секретним.

Для аналізу таких криптосистем існує багато методів. Але слід враховувати, що при аналізі асиметричних криптосистем всі методи зводяться до розв’язку двох задач різними способами – задачі дискретного логарифмування та задачі факторизації великих чисел. Розглянемо цей вид криптоаналізу у наступних статтях.

У статті [17] автор досліджує паралельний алгоритм дискретного логарифмування методом решета числового поля. При цьому дослідженні виявлено що важливу роль в оптимізації часу обчислення дискретного логарифму грає розмір базису. Якщо розмір базису буде занадто великим, то можна буде легко знайти гладкі числа, але достатньо важко буде здійснювати перевірку на гладкість. Якщо навпаки, розмір базису буде малим, то перевірка на гладкість і Гаусовий виняток буде легко здійснюватися, але складно буде знайти достатню кількість гладких чисел. Також на швидкість виконання обчислень має вплив параметри обчислювальної системи – кількість та продуктивність процесорів, а найголовніше – передавальне довкілля.

В статті [18] представлено проект реалізації алгоритму шифрування Мілера-Рабіна на мові $C\#$, які функціонують швидше стандартного алгоритму на 50%, що обумовлює спрощення роботи при створенні ключей для алгоритмів шифрування типу RSA. В статті [18] досліджено алгоритм перевірки чисел на простоту Міллера-Рабіна, модифікацію якого збільшує швидкість дії реалізованого стандартного алгоритму. Автор представляє програмну реалізацію алгоритму Міллера-Рабіна, яка є ефективнішою порівняно зі стандартним ітераційним алгоритмом. Ці методи сході за функціоналом, але мають різну оцінку структурної складності. Саме тому, алгоритм Міллера-Рабіна є надійним, швидкодіючим, стійким, що підтверджено експериментальним методом дослідження.

В статті [19] автор досліджує деякі елементи теорії чисел і представляє їх використання у сучасних криптосистемах. у статті наведені приклади відомих протоколів та алгоритмів, такі як протокол Діффі-Хеллмана, для створення парного ключа, алгоритми шифрування з відкритим ключем RSA та Ель Гамаля. В історичній частині описано алгоритм Евкліда, який є одним з примітивів в теорії чисел, який використовується в криптографії. Наведено алгоритми електронного підпису RSA та Ель Гамаля. Автор пропонує алгоритм електронного підпису, що базується на білінійному перетворенні, який використовує спрощений вид взаємодії.

Для аналізу симетричних криптосистем існує різноманітні методи, що використовують лінеаризація, різностні характеристики пар текстів, складання систем перевизначених рівнянь. Саме тому в статті [20] досліджено основні поняття побудови та аналізу сучасних криптографічних систем. Для симетричних алгоритмів шифрування представлена можливість застосування методів диференціального та алгебраїчного аналізів. Також для асиметричних криптосистем – алгоритми аналізу, що побудовані на основі методів факторизації та дискретного логарифмування. Автор досліджує окремо підходи до аналізу сучасних функцій хешування. В статті наведені експериментальні дані, які отримано за допомогою реалізованих програм, які

відображають ефективність розроблених алгоритмів. Для паралельних алгоритмів отримано результати, що відображають залежність швидкості обчислень від використаного числа процесорів та способу розподілу даних.

Алгоритми шифрування, які базуються на основі еліптичних кривих та алгебраїчних структур. Якщо необхідно передати секретну інформацію по відкритим каналам зв'язку при наявності у обох сторін зв'язку спільного секретного ключа малого розміру (наприклад, 32, 40 або 56 біт), то шифрування за роздільним ключем небезпечною операцією. Оскільки зловмисник отримує практичну можливість підібрати ключі шляхом перебору. Але ключ даного розміру має зацікавленість у реалізації в протоколах, які використовують симетричну і асиметричну криптографію. Для надійного захисту інформації, яку передають по відкритим каналам зв'язку, в умовах обмеженості ключової бази було запропоновано протокол стійкого шифрування за розділеним ключем малого розміру в групі точок еліптичної кривої. Цей протокол описано у статті [21]. Протокол реалізовано на основі поєднання процедур шифрування та шифрування без ключа за секретним ключем малого розміру (до 56 біт). Для зростання продуктивності процедур даного шифрування виникає потреба у реалізації вказаного протоколу в групі точок еліптичної кривої. Автор представляє два протоколи. У першому протоколі аутентифікація здійснюється без зміни початкового протоколу, що створює обмеження на повторне використання секретного ключа малого розміру. Через це автор пропонує нове функціонування аутентифікації, яка дозволяє прибрати вказане обмеження. Така побудова протоколу з використанням обчислень еліптичних кривих, яке здійснюється з модулярною арифметикою, забезпечено механізмом випадкового кодування повідомлень, які зашифровуються точками еліптичних кривих. Через це дешифрування секретної інформації зловмисником закінчується руйнуванням процесу розсекречення.

Наприклад, в статті [22] автор пропонує покращений аналог системи RSA. Така система будується на основі $M_2(Z_n)$ – кільці матриць порядку два

над кільцем лишків Z_n. Також автор пропонує ще один аналог алгоритму RSA, в якому ключ дешифрування визначається не за функцією Ейлера, а за періодом мультиплікативної групи. Автор знаходить залежність між величиною порядку групи та значенням функції Ейлера, яка дозволяє зробити висновок про те, що аналогічна система має більшу криптостійкість, тому що для знаходження періоду групи необхідно знати розклад відкритого модуля у добуток двох різних простих чисел, а це є складною задачею.

В статті [23] описано метод комплексного добутку для генерації кривих, що підходять для криптографії. На основі цього розроблено універсальний алгоритм. Автор досліджує алгоритм знаходження кінцевої групи, що підходить для криптографії, використовуючи теорію еліптичних кривих з комплексним добутком.

Автор ставив за мету надати опис методу комплексного добутку для генерації кривих та розробити відповідний алгоритм. Цей алгоритм має таку структуру: знаходження класів ізоморфних абелевих многовидів; знаходження класових поліномів; визначення рівняння кривої. Найбільшу складність представляє факторизація класового поліному.

У роботі [24] автор пропонує невелике представлення криптографії, яка заснована на групах – сучасного напрямку, де головними об’єктами є абстрактні нескінчені групи. Автор ставить за мету побудову на групах криптографічних примітивів, систем і протоколів. Саме в цьому напрямку дослідження використано теорію груп, теорію складності і теорію обчислень. Автор звертає увагу на використання нерозв’язаних і складнорозв’язаних алгоритмічних проблем теорії груп у якості зазначеної побудови. Автор обговорює аспекти складності алгоритмічних проблем і пов’язаних з ними проблемами пошуку. Автор описує універсальність діофантової мови у криптографії, за допомогою якої будується одностороння функція, що грає велику роль у криптостійкості.

У статті [25] автор пропонує геометричну модель, яка призначена для уніфіційованого підрахунку числа елементів кінцевих множин, які визначені в термінах класів порівнянь за попрано простим модулем.

Автор звертає увагу на систематичне застосування теорії кінцевих кілець і модулей лінійних форм над кінцевими кільцями. У процесі розв'язку задач криптографії теоретичне дослідження цієї загальної комбінаторної схеми є актуальним як для комбінаторного аналізу, так і для прикладного аналізу. конструкція, яку запропонував автор, є криптостійкою.

У статті [26] автор наводить приклад шифру інформації, який реалізується на використанні розробленого алгоритму еліптичного шифрування. Таким чином автор досліджує алгоритм криптографії, який будується на еліптичних кривих з використанням модулярної арифметики. Такий алгоритм дозволяє здійснити операції шифрування та дешифрування. Також автор порівнює даний алгоритм з алгоритмом RSA, наголошуючи на те, що досліджуваний алгоритм є більш швидкодіючим та зашифровує велику кількість інформації.

У статті [27] представлено поміхостійка модулярна криптосистема, яка функціонує в кільці Z_p додатних цілих чисел за модулем р. Запропоновано алгоритм розширення системи основ криптосистеми. Представлена оцінка поміхостійкості криптосистеми за відношенням до традиційним методам поміхо- та крипто захисту, в яких шифрування та дешифрування здійснюється незалежно, з використанням різних алгоритмічних методів. Автор пропонує метод об'єднання процедур шифрування та дешифрування для захисту даних від впливу відмінностей різноманітного походження при передачі їх по відкритим каналам зв'язку.

Криптосистеми на основі еліптичних кривих з'явились після праці Ніла Кобліца. Такі криптосистеми є криптостійкими порівняно з іншими асиметричними алгоритмами зі значно меншою довжиною ключа. В статті [28] автор виказує думку про неможливість відмови від досліджень можливості побудови криптосистеми на основі еліптичної кривої, коли її

коефіцієнти належать кінцевому ірраціональному полю та обчислення здійснюються за ірраціональним модулем. Якщо отримано розв'язок даної задачі, то ми отримаємо криптосистему з більш високим порядком групи точок еліптичної кривої для заданого модуля порівняно з еліптичною кривою над кінцевим полем. Саме про це докладно розповідає автор у статті [28]. Автор розробляє криптосистему над кінцевим ірраціональним полем, використовуючи властивості числової системи Бергнмана з ірраціональною основою та властивості цілочисельних послідовностей Фібоначі.

Функції хешування. У 1989 р. Р. Меркль та І. Дамгорд запропонували ітеративний принцип побудови функції хешування. Такий спосіб зводить задачу до побудови хеш-функції на множині повідомлень різної довжини до задачі побудови відображення, яке здійснюється на множині фіксованої кінцевої довжини. На базі властивостей криптографічних функцій хешування виділяють 3 типи атак: атака на знаходження колізій; атака знаходження першого прообразу; атака знаходження другого прообразу. Для реалізації даних атак може бути застосовані різні методи: методи, незалежні від алгоритму перетворення; методи, що базуються на слабкості алгоритму перетворення хеш-функції.

Наприклад у статті [29] автор досліджує основні підходи до аналізу сучасних функцій хешування з використанням методу диференційного криптоаналізу на прикладі спрощених версій функцій SHA. Виявлено що хеш-функції розв'язують 2 основні задачі криптографії: побудова систем контролю цілісності даних при їх передачі або зберігання; аутентифікація даних. Автор пропонує нові методи аналізу, які розраховані на широкі класи хешування, а саме метод диференційного криптоаналізу.

2.3 Висновки за розділом 2

Отже, у розділі 2 ми дослідили основні класичні шифри криптографії, які використовували модулярну арифметику в основі алгоритму шифрування. Ми дослідили історію появи та подальший розвиток використання модулярної арифметики у 15-19 ст.

Також у розділі 2 ми дослідили сучасний стан криптографії та криптоаналізу, що використовують модулярну арифметику. Провели аналіз сучасного стану криптографії. Виділили основні методи та типи шифрування, які є розповсюдженими серед криптоаналітиків. Надали короткий опис кожному з типів алгоритмів шифрування, наведених у пункті 2.2.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ КРИПТОГРАФІЇ

3.1 Сучасні алгоритми криптографії

3.1.1 Гамування

В основі даних систем шифрування полягає метод «накладання» ключової послідовності – гамми на відкритий текст. «Накладання» створюється за допомогою додавання знаків (літер) або різниці за модулем. Данні шифросистеми відносяться до багатоалфавітних систем заміни. Шифри гамування мають цілий ряд особливостей. В силу простоти технічної реалізації ці шифри отримали широке розповсюдження.

Для дослідження шифру гамування звернемося до підручників [30, 31], де описано принцип дії даного шифру.

Історично перший шифр гамування був схожий на шифр Віженера, однак без використання таблиці Віженера.

Шифр табличного гамування в алфавіті $G = \{a_1, \dots, a_n\}$ визначається довільним латинським квадратом T на G і способом отримання послідовності літер з G, яка отримала назву гамма шифру (див. рис. 3.1). Буква a_i відкритого тексту під дією знаку гамми a_j переходить у літеру a_k тексту шифрування, яка знаходиться у $j-$ому рядку та $i-$ому стовпці квадрату T (мається на увазі що рядки в T мають номери у відповідності з порядком послідовності літер в алфавіті G) [30].

З алгебраїчної точки зору літера a_k є результатом застосування до літер a_i та a_j квазігруповою операцією *, табличним заданням якої є латинський квадрат T:

$$a_k = a_i \cdot a_j. \tag{3.1}$$

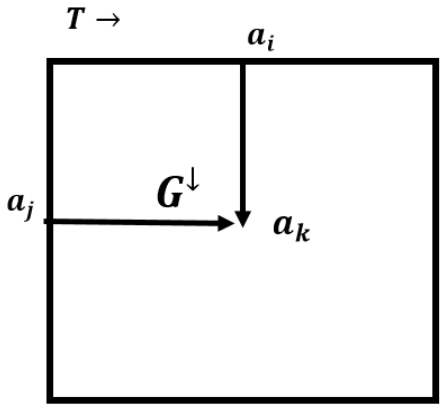

Рисунок 3.1 – Латинський квадрат L

У випадку шифру Віженера квазігрупа $(A,*)$ є групою $(Z_n,+)$. При цьому рівняння шифрування представляється у виді [30]:

$$b_i = (a_i + \gamma_i)\ mod\ n, \tag{3.2}$$

а $\{\gamma_i\}$ представляє собою періодичну послідовність, що утворюється повторенням деякого ключового слова [30].

Поряд з додаванням використовується й віднімання знаків гамми. Відповідні рівняння шифрування можуть бути записані мовою теорії чисел такими формулами [30]:

$$b_i = (a_i - \gamma_i)\ mod\ n \tag{3.3}$$

або

$$b_i = (\gamma_i - a_i)\ mod\ n. \tag{3.4}$$

Шифри гамування з рівняннями шифрування (3.2) – (3.4) зазвичай називають шифрами модульного гамування.

Якщо у якості квазігрупової операції * на множині 5-вимірних двійкових векторів використовується операція покоординатного додавання за модулем 2 [30]:

$$b_i = a_i + \gamma_i, \tag{3.5}$$

то отримаємо шифр Вернама.

Зауваження [30]. Як шифр заміни, довільний шифр гаммування має наступну інтерпретацію.

З j-м рядком латинського квадрату L ($j = \overline{1,n}$) можна поєднати підстановку g_j (зсув на a_j):

$$g_j = \begin{pmatrix} a_1 & a_2 \dots & a_n \\ a_1 \cdot a_j & a_2 \cdot a_j \dots & a_n \cdot a_j \end{pmatrix}$$

з симетричної групи $S(A)$. Нехай

$$R(A) = \{g_j : j = \overline{1,n}\}.$$

Тоді у кожному такті шифрування знак відкритого тексту замінюється по одній з підстановок з $R(A)$. Розподільником такого n-алфавітного шифру заміни є сама гамма шифру. Іноді $R(A)$ – це група або суміжний клас за деяким класом підгрупи $S(A)$. У таких випадках шифр табличного гамування називається груповим. Довільний шифр табличного гамування не досить зручний для практичної реалізації. Більш зручним є саме груповий шифр, до яких відноситься шифр модульного гамування [30].

Для того, щоб продемонструвати дію шифру, наведемо приклад.

Приклад 3.1 Шифрування буде здійснюватися на основі українського алфавіту ($N = 33$), відкрите повідомлення – «Математика», гама – «циклізація» (таблиця 3.1).

Таблиця 3.1 – Вихідний текст гами

А	**Б**	**В**	**Г**	**Г**	**Д**	**Е**	**Є**	**Ж**	**З**	**И**	**І**	**Ї**	**Й**	**К**	**Л**	**М**
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Н	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Ю**	**Я**
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Тобто при заміні символів на числа літера «М» представлено як «16», «А» – «0», і т.д.

Результат такого алгоритму шифрування ми наведемо за допомогою таблиці 3.2.

Приклад 3.2 Застосуємо процедуру гамування до вихідного тексту «Будь». Наведемо приклад шифрування за допомогою таблиці 3.3.

При такому методі шифрування символи вихідного тексту і гами перетворюються у двійковий код, після чого відповідні розряди складаються за модулем 2. Замість додавання за модулем 2 таке шифрування можна здійснювати на основі інших логічних операцій. Наприклад операція перетворення за правилами логіки еквівалентності та нееквівалентності.

Таблиця 3.2 – Шифр гами

СИМВОЛ											
	Відкритого повідомлення, P_i	М	А	Т	Е	М	А	Т	И	К	А
		16	0	22	6	16	0	22	10	14	0
	Гами, K_i	Ц	И	К	Л	І	З	А	Ц	І	Я
		26	10	14	15	11	9	0	26	11	32
	Шифрограми, C_i	З	И	Г	С	Ч	З	Т	Г	Х	Я
		9	10	3	21	27	9	22	3	25	32

Таблиця 3.3 – Шифр гами

Вихідний текст	Б 010010	У 100000	Д 110010	Ь 100000
Знаки гами	7 000111	1 000001	8 001000	2 000010
Зашифрований текст	010101	1000001	111010	100010

Гамування як один з ефективних методів шифрування є стійким до криптоатак. Це обумовлюється властивістю гами – тривалістю періоду і рівномірністю статистичних характеристик [31].

Гамування розділяють на два види – гамування з кінцевої гами та нескінченої гами. Якщо при шифруванні ми маємо гарні статистичні властивості гами, то стійкість шифру буде залежати від довжини періоду гами. При цьому, якщо довжина вихідного тексту менша за довжину гами, то шифр є абсолютно стійким, тобто його не можна заламати, застосовуючи статистичну обробку вихідного тексту. Але при наявності деякої додаткової інформації вихідний текст може бути частково зламаним, або, навіть, повністю [31].

Режим шифрування одноразового гамування одним ключем двох видів відкритого тексту реалізується наступним чином [31] (рисунок 3.2).

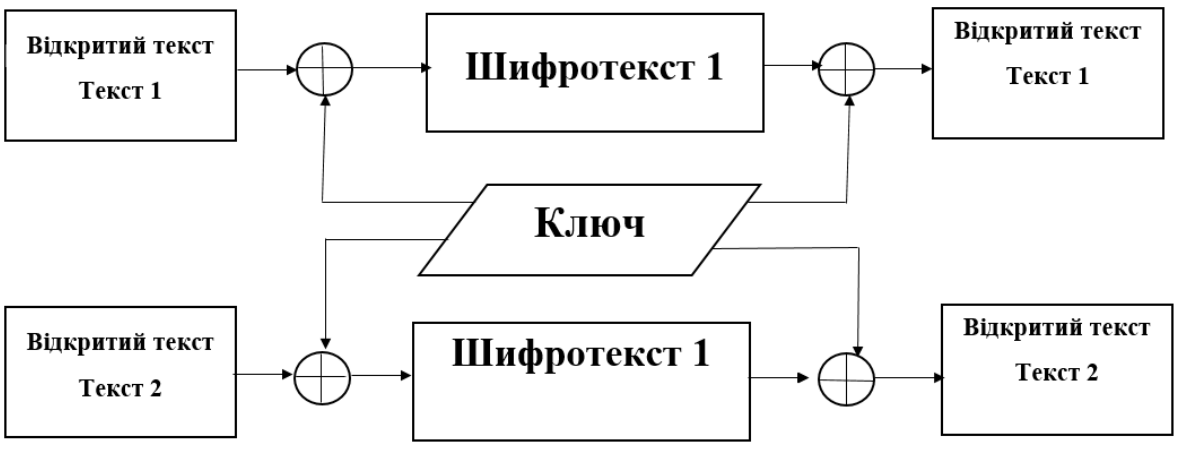

Рисунок 3.2 – Загальна схема шифрування двох різних текстів одним ключем

За допомогою формул режиму одноразового гамування отримаємо шифротексти обох повідомлень [31]:

$$C_i = (P_i + K_i)\ (mod\ N),$$
$$P_i = (C_i + N - K_i)(mod\ N), \qquad (3.6)$$

де C_i, P_i – і-ий символ відкритого та шифрованого повідомлення; N – кількість символів у алфавіті; K_i – і-ий символ гами (ключа).

Задача знаходження відкритого тексту за відомим шифротекстом двох повідомлень, при шифруванні яких використано один ключ, може бути розв’язана, використовуючи формули (3.6).

3.1.2 Робота скремблера

Скремблер – програмна або апаратна реалізація алгоритму, яка дозволяє шифрувати побітовно неперервний потік інформації [32].

На сьогоднішній день існує багато систем скремблерування, які мають відмінності від скремблер-оригіналу. Шифрування за допомогою скремблера є дуже складним процесом. В таких системах початковий сигнал перетворюється у цифрову форму, після чого здійснюють шифрування даних і відправляють їх. Маючи схожість з системами асиметричного шифрування, скремблери є більш стійкими до криптоатак. Такі системи вважаються надійними у роботі з секретною інформацією.

Розглянемо зсувний реєстр з зворотним зв’язком (Linear Feedback Shift Register, скорочено – LFSR) – логічний пристрій, схему представлено на рис. 3.3 [32].

Зсувний реєстр представляє собою послідовність біт. Кількість біт визначається довжиною зсувного реєстру. Якщо довжина дорівнює n біт, то реєстр називається n-бітним зсувним реєстром. Коли потрібно вилучити біт, всі біти зсувного реєстру зсуваються праворуч на 1 позицію. Новий крайній

лівий біт є функцією всіх бітів реєстру, які залишились. На виході зсувного реєстру виявляється молодший значущий біт.

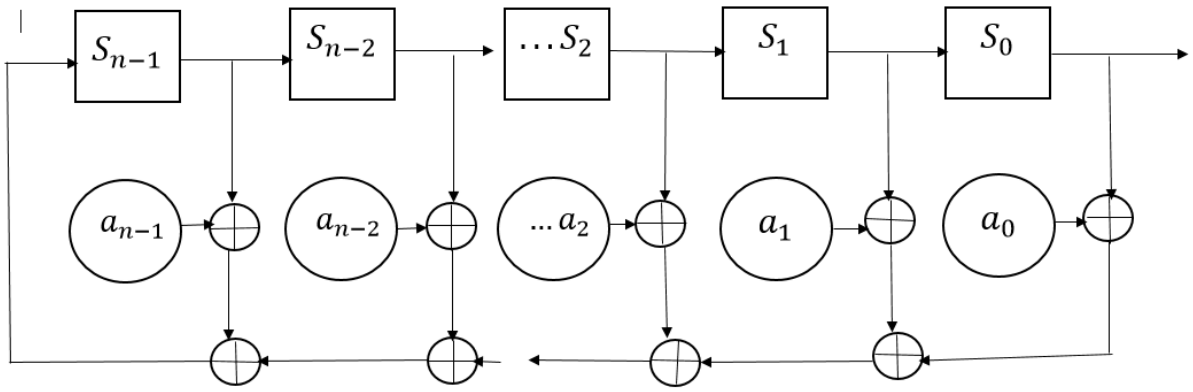

Рисунок 3.3 – Схема LFSR

LFSR складається з n комірок пам'яті, двійковий стан яких у момент часу $t = 0,1, \dots$ характеризується значеннями $S_0(t), S_1(t), \dots, S_{n-1}(t) \in A = \{0,1\}$. Вихід комірок пам'яті пов'язані не тільки послідовно один з одним, але й з суматорами $\oplus$ у відповідності з коефіцієнтами передачі $a_0, a_1, \dots, a_{n-1} \in A$: якщо $a_i = 1$, то значення $S_i(t)$ і-й комірки передається на один з виходів і-го суматора; якщо ж $a_i = 0$, то така передача відсутня. Коефіцієнти передачі можна задати за допомогою полінома [32]:

$$f(x) = x^n + a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \dots + a_2 \cdot x^2 + a_1 \cdot x + a_0.$$

Стан LFSR у даний момент часу t задається двійковим n-вектор-стовбцем [32]:

$$S(t) = \big(S_{n-1}(t), \dots, S_0(t)\big)'.$$

Зміст комірок LFSR з плином часу змінюються наступним чином, визначаючи тим самим динаміку станів LFSR [32]:

$$S_i(t+1) = \begin{cases} S_{i+1}(t), & \text{якщо } i = \overline{0, n-2}, \\ \sum_{j=0}^{n-1} a_j S_j(t), & \text{якщо } i = n-1. \end{cases}$$

Поточне значення нульової комірки реєстру використовується у якості елементів що породжуються LFSR двійкової псевдовипадкової послідовності $S_y = S_0(t)$[32].

Дана послідовність вилучених біт має задовольняти наступним умовам: бути збалансованою, тобто для кожного інтервалу послідовності кількість двійкових одиниць повинна відмінюватися від числа двійкових нулів не більше, ніж на декілька відсотків від їх спільної кількості на інтервалі; бути циклічною, а саме неперервною послідовністю однакових двійкових чисел. Поява іншої двійкової цифри породжує новий цикл. Довжина циклу дорівнює кількості однакових цифр у ньому. Необхідно, щоб половина всіх «смужок» (підряд ідентичних компонентів послідовності) мала довжину 1, одна чверть – довжину 2, одна восьма – довжину 3, і т.і. Умова кореляції – якщо частина послідовності і її циклічно зсунута копія порівнюються за елементами, треба щоб число збігів відмінювалось від числа не збігів не більш, ніж на декілька відсотків від довжини послідовності [32].

Якщо скремблер виконує велику кількість роботи, як наслідок, може виникнути зациклення. Коли виконується визначена кількість тактів у комірках скремблера, послідовність шифру починає циклічно повторюватися з фіксованим періодом.

Ця проблема досить не має розв'язку, так як в N розрядах скремблера не може містити більш ніж 2^N комбінацій біт. З цього випливає, що максимум через $2^N - 1$ ітерацій циклу здійсниться повторення комбінації. Послідовність біт, що генерується таким скремблером, має назву послідовності найбільшої довжини [32].

Для того щоб побудувати N-розрядний скремблер, який утворює послідовність найбільшої довжини, треба брати за основу примітивні многочлени. Примітивний (базовий) многочлен степеня n за модулем 2 – це

незведений многочлен, який є дільником $x^{2^n+1}+1$, але не є дільником x^h+1 для всіх h, на які ділиться 2^n-1. Незведений многочлен степеня n неможливо представити у виді добутку ніяких інших многочленів, крім нього самого та одиничного.

Такий примітивний многочлен степеня n представляють у двійковому виді, потім прибирають одиницю, яка відповідає найбільшому розряду.

Наведемо приклад 5-розрядного скремблера, який генерує послідовність з періодом $T=5: x^7+x^6+x^2$. Нехай початкове значення стану буде дорівнювати $(79)_{10}=(1001111)_2$. Для цього зсувного реєстру новий біт генерується за наступною схемою [32] (рисунок 3.4):

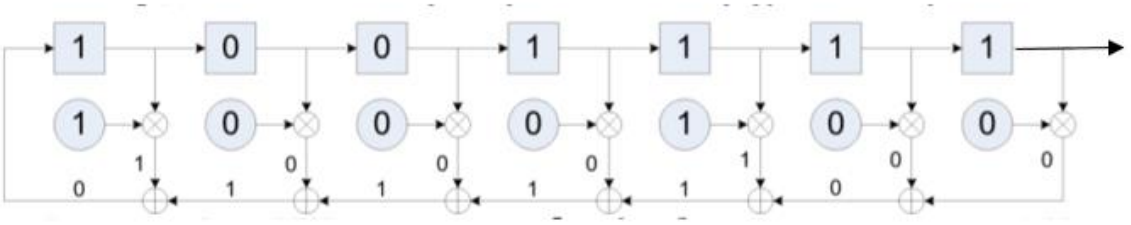

Рисунок 3.4 – Схема LFSR для многочлена $x^7+x^6+x^2$ при початковому стані $(1001111)_2$

3.1.3 Приклади практичної реалізації

На основі пункту 3.1.1 та 3.1.2 наведемо приклади практичної реалізації роботи шифру гамування.

В загальному вигляді структурну схему алгоритму шифрування, що базується на гамуванні можна показати наступним чином [32] (рисунок 3.5).

З рисунку 3.5, ми бачимо, що шифрування вихідної інформації здійснюється поетапно: перший блок даних шифрується ключем K1, відповідно n-й блок – ключем Kn . Саме від надійності ключа і криптографічної стійкості шифру залежить ефективність дії шифру. Створення стійкого шифру гамування полягає в забезпеченні таких властивостей: послідовність гами має бути повністю випадковою, а також

неможливість відкриття невідомих частин гами і ключа за відомими. Результат шифрування буде складним для відкриття в тому випадку, якщо в гамі не будуть повторюватися бітові послідовності. Також важливим є той факт, що коли зловмиснику стає відомим фрагмент вихідного тексту і відповідний йому шифртекст стає легким завдання відновлення всієї послідовності, гамування в такому разі є неефективним. Структурна схема демонструє загальний алгоритм шифрування методом гамування, який можна вдосконалювати шляхом розробки нових модифікацій шифру. Шифри, що базуються на принципі гамування характеризуються надійністю, тому їх широке використання цілком очевидне. На даний час розроблено багато варіантів шифру гамування, наприклад, режим гамування зі зворотнім зв'язком. Суть даного методу полягає в сумуванні за модулем два блоку відкритого тексту з зашифрованим попереднім блоком. Шифрування таким методом може бути описане системою виразів.

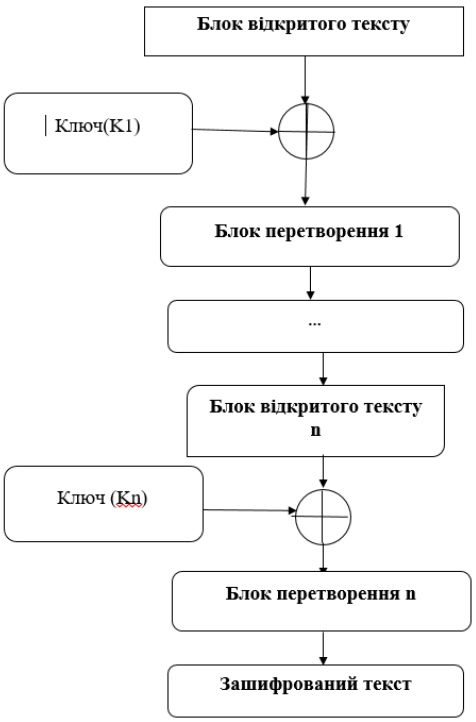

Рисунок 3.5 – Структура алгоритму шифрування

До переваг шифрів гамування можна віднести наступні: висока швидкість шифрування, потоковість шифрування та дешифрування,

збереження розміру інформації при шифруванні. Проте, існують і суттєві недоліки даних шифрів, такі як нестійкість шифру при повторному застосуванні та послідовність доступу до інформації. Більшість відомих до цього часу методів гамування мають ряд недоліків, уникнути яких можна застосовуючи модифікацію шифру гамування. Суть вдосконаленого шифру гамування полягає в шифруванні за допомогою зміненого ключа з кожним наступним блоком інформації.

Приклад 3.3 Розглянемо у вигляді вихідного тексту інформацію, що має представлення у двійковому виді та відповідно до цього ключ виду (табл. 3.4).

Таблиця 3.4 – Вихідний текст та відповідний ключ

Вихідний текст	010000101111 010001000000 010000111110 010001000001 010000111011 010000110000 010000110010 000000001010 010000011010 010001000000 010000111000 010000110010 010000111110 010000111001 000000001010
Ключ	010001010001 010000110011 010000110011 000001110100 000000110100 000000111001 010001000000 010001000000 000001100010 000001111001 010001001010 000001110101 010001000010 000001111000 010001001011

Дешифрування вихідного тексту у двійковому представленні, що здійснилось на основі роботи шифру гамування має вид (таблиця 3.5).

Таблиця 3.5 – Шифрограма

Шифрограма	000001111110 000001110011 000000001101 010000110101 010000001111 010000001001 000001110010 010001001010 010001111000 010000111001 000001110010 010001000111 000001111100 010001000001 010001000001

Приклад практичної реалізації дії шифру гамування за допомогою програми Java наведено у Додатку А.

Розглянемо тепер практичну реалізацію скремблера. Отже, спробуємо побітовно змінити інформацію, що проходить через інформаційний потік.

Приклад 3.4 Розглянемо у вигляді вихідного тексту інформацію, що представлена у двійковому виді та відповідний ключ наступним чином (таблиця 3.6).

Таблиця 3.6 – Вихідний текст та ключ

Вихідний текст	010000101111 010001000000 010000111110 010001000001 010000111011 010000110000 010000110010 000000001010 010000011010 010001000000 010000111000 010000110010 010000111110 010000111001 000000001010
Ключ	010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000 010111100000

Шифрограмму та дешифрування представлено у таблиці 3.7.

Таблиця 3.7 – Шифрограма

Шифрограмма	000111001111 000110100000 000111011110 000110100001 000111011011 000111010000 000111010010 010111101010 000111111010 000110100000 000111011000 000111010010 000111011110 000111011001 010111101010
Дешифрування	010000101111 010001000000 010000111110 010001000001 010000111011 010000110000 010000110010 000000001010 010000011010 010001000000 010000111000 010000110010 010000111110 010000111001 000000001010

Проведемо співставлення двійкового коду та дешифрованого тексту (таблиця 3.8).

Таблиця 3.8 – Шифрограмма та дешифрований текст

Шифрограмма	000111001111 000110100000 000111011110 000110100001 000111011011 000111010000 000111010010 010111101010 000111111010 000110100000 000111011000 000111010010 000111011110 000111011001 010111101010
Дешифрований текст	010000101111 010001000000 010000111110 010001000001 010000111011 010000110000 010000110010 000000001010 010000011010 010001000000 010000111000 010000110010 010000111110 010000111001 000000001010

Всі розрахунки практичної реалізації роботи скремблера за допомогою програми Java представлено у Додатку А.

Дослідимо роботу скремблера за допомогою програмного математичного пакету Maple. Для цього ми скористуємося командою «array»щоб задати масиви даних, з якими ми будемо працювати далі (рис. 3.6).

```
> restart;
> x := array(1..32) : y := array(1..32) : z := array(1..32) :
>
>
  x[1] := 1 : x[2] := 2 : x[3] := 3 : x[4] := 4 : x[5] := 5 : x[6] := 6 : x[7] := 7 : x[8] := 8 :
  x[9] := 9 : x[10] := 10 : x[11] := 11 : x[12] := 12 : x[13] := 13 : x[14] := 14 : x[15] := 15 : x[16] := 16 :
  x[17] := 17 : x[18] := 18 : x[19] := 19 : x[20] := 20 : x[21] := 21 : x[22] := 22 : x[23] := 23 : x[24] := 24 :
  x[25] := 25 : x[26] := 26 : x[27] := 27 : x[28] := 28 : x[29] := 29 : x[30] := 30 : x[31] := 31 : x[32] := 32 :


>
  y[1] := "А" : y[2] := "Б" : y[3] := "В" : y[4] := "Г" : y[5] := "Д" : y[6] := "Е" : y[7] := "Ё" : y[8] := "Ж" :
  y[9] := "З" : y[10] := "И" : y[11] := "Й" : y[12] := "К" : y[13] := "Л" : y[14] := "М" : y[15] := "Н" : y[16] := "О" :
  y[17] := "П" : y[18] := "Р" : y[19] := "С" : y[20] := "Т" : y[21] := "У" : y[22] := "Ф" : y[23] := "Х" : y[24] := "Ц" :
  y[25] := "Ч" : y[26] := "Ш" : y[27] := "Щ" : y[28] := "Ы" : y[29] := "Ь" : y[30] := "Э" : y[31] := "Ю" : y[32] := "Я" :
```

Рисунок 3.6 – Програмна реалізація роботи скремблер в Maple

Далі ми поставимо у відповідність елементи вихідних масиві (рис. 3.7).

```
> for i from 1 to 32 do
      z[i] := convert(x[i], binary) :
  end do:
  print(`x= `,x) : print(` y=`,y) : print(` z=`,z) :
                    x= , [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 ]
y=,
   ["А","Б","В","Г","Д","Е","Ё","Ж","З","И","Й","К","Л","М","Н","О","П","Р","С","Т","У","Ф","Х","Ц","Ч","Ш","Щ","Ы","Ь","Э","Ю",
   "Я"]
z=, [1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, 10101, 10110, 10111, 11000, 11001, 11010, 11011,
   11100, 11101, 11110, 11111, 100000]
```

Рисунок 3.7 – Програмна реалізація роботи скремблер в Maple

Після цього задамо вихідний текст для роботи шифру (рис. 3.8).

```
> s := "ПРИВЕТЯМЕРЗЛИКИНАВАЛЕРИЯ";
  CODE := "" :
  v := array(1 .. length(s)/2) :

                                   s := "ПРИВЕТЯМЕРЗЛИКИНАВАЛЕРИЯ"
```

Рисунок 3.8 – Програмна реалізація роботи скремблер в Maple

Далі задамо необхідний цикл, який буду генерувати шифротекст, тобто у відповідність кожній літері кирилиці буде ставити у відповідність (кодувати) символ двійкової системи числення (конвертування) (рис. 3.9).

```
> for i from 1 to length(s)/2 do
      v[i] := s[2·(i − 1) + 1 ..2·i] :
      for j from 1 to 32 do
        if (v[i] = y[j]) then
                    # print(z[j]);
                     CODE := cat(CODE, z[j], " ");
        end if:
      end do:
  end do:
  print(CODE);
```

Рисунок 3.9 – Програмна реалізація роботи скремблер в Maple

Отримаємо такий результат (рис. 3.10).

```
print(CODE);
"10001 10010 1010 11 110 10100 100000 1110 110 10010 1001 1101 1010 1100 1010 1111 1 11 1 1101 110 10010 1010 100000 10001 10010 1010 11 110 10100
 100000 1110 110 10010 1001 1101 1010 1100 1010 1111 1 11 1 1101 110 10010 1010 100000 "
```

Рисунок 3.10 – Програмна реалізація роботи скремблер в Maple

Ми отримали зашифрований вихідний текст, представлений у двійковій системі.

Дану програму можна застосовувати для кодування будь-якого тексту, написаного за допомогою кирилиці. Всі розрахунки практичної реалізації роботи скремблера за допомогою програми Maple представлено у Додатку Б.

3.2 Генерація незведених многочленів, які пов’язані степеневою залежністю коренів

Генерація незведених многочленів є розповсюдженою та непростою задачею, яка застосовується в прикладній математиці. Постановка такої задачі зводиться до пошуку незведених многочленів з властивостями, які можна використовувати для створення ключів при передачі інформації у захищеній криптосистемі.

Такий вид генерації ключів використовують у сучасній криптографії, для покращення криптостійкості шифросистем. У пункті 3.2., ми дослідимо методи і алгоритми, на основі яких базуються генерація незведених многочленів, а саме знаходження нових незведених многочленів з даного незведеного многочлену того самого степеня за умовою, що корені многочленів пов’язані довільними степеневими залежностями. Для цього ми звернемося до методології дослідження автора статті [33], і скористаємося необхідними означеннями, теоремами та твердженнями, які будуть корисними для дослідження.

Означення 3.2 Нехай K/k – кінцеве розширення полів степенів $n, \omega_1, \dots, \omega_n$ – базис K над k, $z \in K, U$ – матриця гомотетії $\hat{z}: K \to K, x \to zx$ в базисі $\omega_1, \dots, \omega_n$. Характеристичним многочленом елементу z називається визначник

$$g(x) = \det(X \cdot E - U),$$

де E – одинична матриця порядку n [33].

Отже, дослідимо перехід $x \to x^3$ та $x \to x^5$. Для цього ми розглянемо мінімальний многочлен g_t елемента $\alpha^t \epsilon F_{q^m}$ над F_q. Обчислення такого многочлену проводять за допомогою характеристичного многочлену f_t цього ж елемента над F_q. Відомо, що $f_t = g_t^r$, де $r = \frac{m}{k}$, де k – степінь многочлена g_t. Оскільки многочлен g_t є незведеним над кільцем $F_q[x]$, число $k = ord(g_t)$, тому число d дорівнює порядку елемента α^t над групою $F_{q^m}^*$, а порядок дорівнює $\frac{e}{\text{НОД}(t,e)}$, отже маємо: числа d, k, r можна визначити [33].

Нехай многочлен $f(x) = \sum_{i=0} a_i x^i$ є незведеним над полем $GF(2)$, степінь якого $degf = N$; $1 = \mu^0, \mu^1, \mu^2, \dots, \mu^{t-1}$ – корені степеня t з одиниці, $\mu^t = 1, \mu \neq 1, t$ непарне, $\mu^{k-1} + \mu^{k-2} + \cdots + \mu + 1 = 0$. Нехай $z = x^t$, f_t-характеристичний многочлен елемента $\beta = \theta^t$, де $f(\theta) = 0$. Для подальшого формулювання загальної формули для обчислення характеристичного многочлену, ми звернемося до статті [33], автор якої пропонує спочатку сформулювати теорему 3.1:

Теорема 3.1 Нехай f – нормований незведений многочлен, який має степінь m з $F_q[x]$. Нехай $\alpha \in F_q m$ – який-небудь корінь цього многочлену, та для $t \in N$ нехай f_t – характеристичний многочлен елемента $\alpha^t \in F_q m$ над F_q. Тоді

$$f_t\,(x^t) = (-1)^{m(t+1)} \prod_{j=1}^{t} f(\mu_j x),$$

де $\mu_1, \dots, \mu_t$ – корені степеня t з одиниці над F_q з урахуванням їх кратності [33].

Приклад 3.5 Розглянемо незведений многочлен $f(x) = x^4 + x + 1 \in F_2[x]$. Для того щоб обчислити f_3, звернемо увагу на те, що корені 3 степеня з одиниці над F_2 сутність 1, ω та ω^2, де ω- корінь многочлену $x^2 + x + 1$, який належить полю F_4. Тоді будемо мати:

$$f_3(x^3) = (-1)^{16} f(x) f(\omega x) f(\omega^2 x) =$$
$$= (x^4 + x + 1)(\omega x^4 + \omega x + 1)(\omega^2 x^4 + \omega^2 x + 1) = x^{12} + x^9 + x^6 + x^3 + 1,$$

Так що ми отримали в результаті $f_3(x) = x^4 + x^3 + x^2 + x + 1$.

Отже за теоремою 3.1 ми можемо записати загальну формулу для обчислення характеристичного многочлену [33]:

$$f_t(x^t) = (-1)^{N(t-1)} \prod_{j=0}^{N} f(\mu^j x).$$

Загальну формулу $f_t(x^t) = (-1)^{N(t-1)} \prod_{j=0}^{N} f(\omega^j x)$ можна використовувати для будь-якого поля $GF(q)$. Так як поле $GF(2)$ є базовою основою у більшості алгоритмів шифрування, наприклад $RSA, DES, RC4,$ та еліптичної криптографії, ми дослідили метод зведення для поля $GF(2)$ для викладання комбінаторного підходу та явного обчислення коефіцієнтів шляхом зведення подібних [33].

У формулі для b_r немає множника μ, отже

$$\mu^{0\,i+1j+\cdots+(t-1)k} = 1. \qquad (3.7)$$

Формула (3.7) можлива тільки тоді, коли

$$0i_0 + 1i_1 + \cdots + (t-1)i_{t-1} \equiv 0(mod t), b_r \in GF(2),$$

після зведення подібних, отримаємо [33]:

$$b_r = \sum_{\substack{0n_0+1n_1+\cdots+Nn_N=tr \\ n_0+n_1+\cdots+n_N=t}} a_0^{n_0} a_1^{n_1} \dots a_N^{n_N} \cdot \varepsilon_{n_0 n_1 \dots n_N}, \qquad (3.8)$$

де $\varepsilon_{n_0 n_1 \dots n_N} \in GF(2) = \{0,1\}$.

Отже, треба дізнатись, саме яких доданків у (3.8) не буде, тобто для яких індексів $\varepsilon_{n_0 n_1 \dots n_N} = 0$. Для цього ми дослідимо процес зведення подібних доданків:

Генерація незведених многочленів за допомогою $x \to x^3$.

Розглянемо многочлен виду [33]:

$$b_r = \sum_{i_0+i_1+i_2=3r} a_{i_0} a_{i_1} a_{i_2} \cdot \mu^{0\cdot i_0+1\cdot i_1+2\cdot i_2},$$

де $\mu^3 = 1, \mu \neq 1, \mu^2 + \mu + 1 = 0$.

Будемо досліджувати декілька випадків, у яких буде різна кількість елементів на множині $\{i_0, i_1, i_2\}$.

Якщо $|\{i_0, i_1, i_2\}| = 1$, тобто $i_0 = i_1 = i_2 = r, 0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2 = 3 \cdot r$, так що $\mu^{0\cdot i_0+1\cdot i_1+2\cdot i_2} = \mu^{3r} = 1$, тому $3 \cdot r \equiv 0 (mod 3)$, це той випадок, при якому можливе внесення доданку a_r^3 у суму для коефіцієнта b_r [33].

Якщо $|\{i_0, i_1, i_2\}| = 2$, то двоє індексів з трьох однакові, але є відмінними від третього. Відзначимо індекси, які співпадають через v, а третій через w. Таким чином ми дослідимо вхід доданку $a_v^2 a_w$, при $i_0 + i_1 + i_2 = 2 \cdot k + l = 3 \cdot r$ [33]. Розподіл $\{v, w\}$ за індексами $\{i_0, i_1, i_2\}$ можливий лише зі списку у таблиці 3.9.

Таблиця 3.9 – Розподіл $\{v, w\}$ за індексами $\{i_0, i_1, i_2\}$

i_0	i_1	i_2	$0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2$	$\mu^{0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2}$
v	v	w	$v + 2 \cdot w$	$\mu^{v+2 \cdot w} = \mu^{3v} \cdot \mu^{2(w-v)} = \mu^{2(w-v)}$
v	w	v	$w + 2 \cdot v$	$\mu^{w+2 \cdot v} = \mu^{3v} \cdot \mu^{(w-v)} = \mu^{(w-v)}$
w	v	v	$3 \cdot v$	$\mu^{3v} = 1$

З таблиці 3.9, ми можемо отримати доданки при зведенні подібних для $a_v^2 a_w$ [33]:

$$\sum_{\substack{i_0=i_1=v, i_2=w \\ i_0=i_2=v, i_1=w \\ i_1=i_2=v, i_0=w}} a_{i_0} a_{i_1} a_{i_2} \cdot \mu^{0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2} = a_v^2 a_w . \left[\mu^{2(w-v)} + \mu^{(w-v)} + 1\right]. \quad (3.9)$$

З цього списку можливі два варіанти: коли v порівняно з w, і, навпаки, коли вони непорівняні.

Розглянемо перший варіант, коли v порівняно з w: $l - k \equiv 0 \ (mod\ 3)$. З цього випливає, що $\mu^{2(w-v)} + \mu^{(w-v)} + 1 = 1 + 1 + 1 = 3(mod 2) = 1$. Дослідимо за модулем 2, так як $\mu \in GF(2)$. Це означає, що доданок $a_v^2 a_w$ буде входити до складу суми коефіцієнта b_r за умовою що $v \equiv w \ (mod\ 3)$ [33].

Тепер розглянемо інший варіант, коли v непорівнянна з w, тобто $w - v \neq 0 \ (mod\ 3)$. Через те, що будь-яка степінь $z = \mu^{(w-v)}$ задовольняє рівнянню $z^2 + z + 1 = 0$, то вираз у квадратних дужках (3.1) буде дорівнювати 0, тобто $\mu^{2(w-v)} + \mu^{(w-v)} + 1 = 0$, при $w - v \neq 0 (mod\ 3)$. Однак їх непорівнянність неможлива, так як з умови $2v + w = 3r \equiv 0 (mod\ 3)$ випливає, що $w \equiv -2v \equiv v (mod\ 3)$, тому $-2 \equiv 1 \ (mod\ 3)$. Отже, доданок $a_v^2 a_w$ не буде входити до складу суми коефіцієнту b_r за умовою $w - v \neq 0 (mod\ 3)$[33].

Отже ми маємо, що доданок $a_k^2 a_l$ буде входити до складу коефіцієнту b_r тільки за умовами $w \equiv v (mod\ 3),\ 2v + w = 3r$.

Якщо $|\{i_0, i_1, i_2\}| = 3$, то всі індекси мають відмінність: $i_0 \neq i_1 \neq i_2$. Розглянемо випадок коли

$$\{i_0, i_1, i_2\} = \{v, w, s\}, v + w + s = 3 \cdot r \qquad (3.10),$$

так що ми будемо доданок $a_v a_w a_s$ [33]. Розподіл v, w, s за індексами i_0, i_1, i_2 пропонує всього 6 перестановок на трьох елементах, інформацію про яких, ми можемо побачити у таблиці 3.10.

Таблиця 3.10 – Розподіл v, w, s за індексами i_0, i_1, i_2

i_0	i_1	i_2	$i_0 + i_1 + i_2 = 3 \cdot r$	$0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2$	$\mu^{0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2}$
v	w	s	$v + w + s = 3r$	$w + 2s$	$\mu^{(w-v)+2(s-v)}$
v	s	w	$v + s + w = 3r$	$s + 2w$	$\mu^{3v}\mu^{(s-v)+2(w-v)}$
w	v	s	$w + v + s = 3r$	$v + 2s$	$\mu^{3v}\mu^{2(s-v)}$
w	s	v	$w + s + v = 3r$	$s + 2v$	$\mu^{3v}\mu^{(s-v)}$
s	v	w	$s + v + w = 3r$	$v + 2w$	$\mu^{3v}\mu^{2(w-v)}$
s	w	v	$s + w + v = 3r$	$w + 2v$	$\mu^{3v}\mu^{(w-v)}$

З цього списку при зведення подібних для $a_v a_w a_s$ отримаємо наступний вираз [33]:

$$a_v a_w a_s \left[\mu^{(w-v)+2(s-v)} + \mu^{(s-v)+2(w-v)} + \mu^{2(s-v)} + \mu^{(s-v)} + \mu^{2(w-v)} + \mu^{(w-v)}\right].$$

Звернемо увагу на те що з (3.10) випливає наступна умова [33]:

$$(v - w) + (w - v) + (s - v) = (w - v) + (s - v) \equiv 0(\, mod\ 3). \qquad (3.11).$$

Це означає що $(w - v) \equiv -(s - v)(mod\ 3)$ та $(w - v) \equiv 2(s - -v)(mod\ 3)$. Виходячи з цього, ми можемо зробити такий висновок, що існує

варіанти коли $(s-v) \equiv 0$, та коли $(s-v)$ порівняно з нулем. Розглянемо більш детально саме ці два варіанти [33]:

а) дано $(s-v) \equiv 0(mod\ 3)$. Отже $s \equiv v(mod\ 3)$, далі отримаємо що $w \equiv v(mod\ 3)$. Після цього можемо зробити висновок що $s \equiv v \equiv w(mod\ 3)$. Через те, о всі доданки дорівнюють одиниці, ми отримаємо наступну формулу [33]:

$$a_k a_l a_m\left[\omega^{(l-k)+2(m-k)} + \omega^{(m-k)+2(l-k)} + \omega^{2(m-k)} + \omega^{(m-k)} + \omega^{2(l-k)} + \omega^{(l-k)}\right]$$
$$a_k a_l a_m \cdot 6 \equiv 0\ (mod\ 2). \qquad (3.12)$$

Отже, згідно з (3.12), якщо $s \equiv w \equiv v(mod\ 3)$, то сума коефіцієнтів b_r не містить доданок $a_v a_w a_s$.

б) дано $m-k \neq 0\ (mod\ 3)$. Отже, маємо:

$$v \neq s\ (mod\ 3), v \neq w(mod\ 3). \qquad (3.13)$$

Вираз (3.13) будемо спрощувати для подальшого дослідження:

$$(w-v) + 2(s-v) = 2(s-v) + 2(s-v) = (s-v)(mod\ 3),$$
$$(s-v) + 2(w-v) = (s-v) + (s-v) = (w-v)(mod\ 3). \qquad (3.14)$$

Згідно з формулою (3.14) будемо мати [33]:

$$a_v a_w a_s\left[\mu^{(w-v)+2(s-v)} + \mu^{(s-v)+2(w-v)} + \mu^{2(s-v)} + \mu^{(s-v)} + \mu^{2(w-v)} + \mu^{(w-v)}\right],$$
$$a_v a_w a_s\left[\mu^{2(s-v)} + \mu^{2(w-v)}\right] = a_v a_w a_s\left[\mu^{2(w-v)} + \mu^{(w-v)}\right]. \qquad (3.15)$$

Так як будь-яка степінь $z = \mu^{w-v}$ задовольняє рівнянню $z^2 + z = 1$, то вираз у квадратних дужках (3.15) не порівняно з нулем, а отже сума коефіцієнтів b_r не буде містити доданок $a_v a_w a_s$, за умовою що $v \neq s(mod\ 3)$.

Також можна зробити висновок про те що $s \neq w(mod\ 3\ )$ за умовами що $(-v) \equiv 2(s-v)(mod\ 3)$ та умов $v \neq s(mod\ 3)$ та $v \neq w(mod\ 3)$[34]. Тому, якщо ми розглянемо випадок коли, $s \equiv v(mod\ 3)$. Здійснимо заміну індексу s на w для виразу $(w-v) \equiv 2(s-v)(mod\ 3)$. Після цієї заміни, ми отримаємо $s - v \equiv 0\ (mod\ 3)$ та $s \equiv v(mod\ 3)$, що заперечує тому що $v \neq s\ (mod\ 3)$.

Тепер дослідимо генерацію незведених многочленів, який здійснюється за допомогою переходу $x \to x^5$. Будемо мати [33]:

$$b_r = \sum_{i_0+i_1+i_2+i_3+i_4=5r} a_{i_0}\, a_{i_1} a_{i_2} a_{i_3} a_{i_4} \cdot \mu^{0\cdot i_0+1\cdot i_1+2\cdot i_2+3\cdot i_3+4\cdot i_4}, \quad (3.16),$$

де $\mu^5 = 1, \mu \neq 1, \mu^4 + \mu^3 + \mu^2 + \mu + 1 = 0$.

Дослідимо 5 випадків, які будуть залежати від кількості різних елементів у наборі індексів $\{i_0, i_1, i_2, i_3, i_4\}$.

Розглянемо випадок коли $|\{i_0, i_1, i_2, i_3, i_4\}| = 1$ [34], отримаємо наступне: $i_0 = i_1 = i_2 = i_3 = i_4 = r$, $0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2 + 3 \cdot i_3 + 4 \cdot i_4 = (1+2+3+4) \cdot r = 10 \cdot r \equiv 0(mod\ 5), \mu^{10r} = 1$. В такому випадку сума коефіцієнтів b_r буде містити єдиний доданок a_r^5 .

Якщо $|\{i_0, i_1, i_2, i_3, i_4\}| = 2$, то будемо мати чотири індексів, що дорівнюють v, але мають відмінності від п'ятого w, або будемо мати три індексів, що дорівнюють v , але будуть відмінні від двох однакових індексів w. Отже перейдемо до двох варіантів перетворень суми коефіцієнтів b_r, в яких буде присутніми доданки $a_v^4 a_w$ та $a_v^3 a_w^2$ відповідно до зазначених двох варіантів [33].

У першому варіанті розглянемо доданок $a_v^4 a_w$. Сума індексів цього доданку $i_0 + i_1 + i_2 + i_3 + i_4 = 4v + w \equiv 5 \cdot r(mod\ 5)$, звідси слідує що $(4v + w) - 5v = (4v - 4v) + (w - v) \equiv 0(mod\ 5)$, тому $w \equiv v\ (mod\ 5)$. Так як, $i_0 \equiv i_1 \equiv i_2 \equiv i_3 \equiv i_4 \equiv v \equiv w\ (mod\ 5)$, звідки ми маємо $0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2 + 3 \cdot i_3 + 4 \cdot i_4 = (1+2+3+4) \cdot v = 10 \cdot v \equiv 0(mod\ 5)$, $\mu^{10v} = 1$ [33]. Числа v та w можна розподілити за індексами i_0, i_1, i_2, i_3, i_4 таким чином, що

ми отримаємо п’ять способів комбінацій: $(kkkkl), (kkklk), (kklkk), (klkkk)$ та $(lkkkk)$. При зведенні подібних ми отримаємо що $5 \cdot a_v^4 a_w = a_v^4 a_w \, (mod\ 2)$. Тому ми можемо зробити висновок, що доданок $a_v^4 a_w$ буде належати сумі коефіцієнту b_r, якщо $w \equiv v(mod\ 5), 4v + w = 5r$. На цьому етапі дослідження доданку $a_v^4 a_w$ завершено.

Тепер розглянемо другий варіант. Обчислення при $a_v^3 a_w^2$ призводять до [33]: $i_0 + i_1 + i_2 + i_3 + i_4 = 3v + 2w \equiv 5 \cdot r \equiv 0(mod\ 5)$, звідки $(3v + 2w) - 5v = (3v - 3v) + (2w - 2w) = 2w - 2v$, тобто $w \equiv v\ (mod\ 5)$ [33]. Отже, індекси при $a_v^3 a_w^2, i_0 \equiv i_1 \equiv i_2 \equiv i_3 \equiv i_4 \equiv v \equiv w(mod\ 5)$. Як наслідок ми маємо $0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2 + 3 \cdot i_3 + 4 \cdot i_4 = (1 + 2 + 3 + 4) \cdot v = 10 \cdot v \equiv 0(mod\ 5),\ \mu^{10v} = 1$. Числа v та w можна утворити $C_5^2 = 10$ способами [33]. Доданок $a_v^3 a_w^2$ не належить сумі коефіцієнту b_r, за умовою що $w \equiv v(mod\ 5), 3v + 2w = 5r$. На цьому етапі дослідження доданку $a_v^3 a_w^2$ завершено.

Розглянемо наступний випадок при $|\{i_0, i_1, i_2, i_3, i_4\}| = 3$ [33]. Будемо мати різні три числа, які відповідно дорівнюють v, w, s. З цього припущення ми дослідимо варіанти, чи будуть доданки $a_v^3 a_w a_s$ та $a_v^2 a_w^2 a_s$ належати сумі коефіцієнту b_r [33].

Розглянемо перший варіант при $a_v^3 a_w a_s$. Будемо мати суму індексів, що дорівнює $i_0 + i_1 + i_2 + i_3 + i_4 = 3v + w + s = 5r$. Таке значення суми індексів є вірним, якщо $(3v + w + s) - 5v = (3v - 3v) + (w - v) + (s - v) \equiv 0(mod\ 5)$. Ми отримали два варіанти, де число$\big(-(s - v)\big) \equiv (mod\ 5)$, або не порівняно. Числа v, w, s можна утворити за індексами $C_5^1 \cdot C_4^1 = 5 \cdot 4 = 20$ способами [33].

Будемо вважати, що

$$\big(-(s - v)\big) \equiv 0(mod\ 5), v \equiv i_0, i_1, i_2, i_3, i_4\ s(mod\ 5),$$

тоді $w - v \equiv 0(mod\ 5), w \equiv v(mod\ 5), v \equiv s \equiv w(mod\ 5)$. При подальших перетвореннях, будемо мати $20 \cdot a_v^3 a_w a_s = 0$. Спираючись на те, що $20 \equiv 0(mod\ 2)$, доданку $a_v^3 a_w a_s$ не буде у сумі коефіцієнту b_r, за умовою що $v \equiv s \equiv w(mod\ 5)$[33].

Припустимо що $\big(-(s - v)\big) \neq 0(mod\ 5)$, будемо мати $s \neq v(mod\ 5)$. Представимо у таблиці 3.11 різниці номерів індексів, які відрізняються від v.

Таблиця 3.11 – Різниці номерів індексів, які є відмінними від v

$s - v$	0	1	2	3	4
0	0	1	2	3	4
1	4	0	1	2	3
2	3	4	0	1	2
3	2	3	4	0	1
4	1	2	3	4	0

В таблиці 3.11 не враховується нулі, які містяться на діагоналі. Множників першого степеню (μ^{s-v}) буде п’ять. Таку саму кількість будуть мати і п’ять множників другого, третього та четвертого степенів. Спираючи на це, ми можемо отримати наступну формулу [33]:

$$a_v^3 a_w a_s \cdot [5 \cdot (\mu^{s-v})^4 + 5 \cdot (\mu^{s-v})^3 + 5 \cdot (\mu^{s-v})^2 + 5 \cdot (\mu^{s-v})^1]\,. \qquad (3.17)$$

У виразі (3.17) будь-яка степінь $z = (\mu^{s-v})$ буде задовольняти рівнянню $z^4 + z^3 + z^2 + z = 1$, тому вираз у квадратних дужках (3.17) буде дорівнювати одиниці. Отже, до суми коефіцієнту b_r, за умовою що $w \neq s \neq v \neq w$, доданок $a_v^3 a_w a_s$ буде входити.

Дослідимо інший варіант, коли ми маємо доданок виду $a_v^2 a_w^2 a_s$. Сума номерів індексів дорівнює $5r$. Звідки будемо мати що

$$(2v + 2w + s) - 5s = 2(v - s) + 2(w - s) + (s - s) \equiv 0(mod\ 5).$$

Отримаємо $(v - s) \equiv -(w - s)(mod\ 5)$. Для подальших перетворень числа v, w, s будемо групувати за індексами i_0, i_1, i_2, i_3, i_4 $C_5^2 \cdot C_3^2 = 10 \cdot 3 = 30$ способами [33].

Припустимо що $\big(-(w - s)\big) \equiv 0(mod\ 5)$[33]. Будемо мати $s - w \equiv 0(mod\ 5), s \equiv w(mod\ 5)$. Виходячи з цього, ми отримаємо що $v - s \equiv 0(mod\ 5), v \equiv s(mod\ 5)$, тобто $v \equiv s \equiv w(mod\ 5)$. зведення доданків призведе нас до $30 \cdot a_v^2 a_w^2 a_s = 0(mod\ 2)$, а це означає що доданку $a_v^2 a_w^2 a_s$ не буде у сумі коефіцієнту b_r, за умовою що $v \equiv s \equiv w(mod\ 5)$.

Припустимо що $(-(w - s)) \neq 0(mod\ 5)$[33]. Щоб дізнатися скільки доданків $\mu^{(v-w)}$ не буде порівняно з нулем, треба розглянути наступну нумерацію індексів: $i_x = v, i_c = v, i_m = w, i_b = w, i_n = s$. Припустимо що $\{x, c, m, b, n\} = \{0,1,2,3,4\}, \{x, c\} \cap \{m, b\} = \emptyset,\ x + c + m + b + n \equiv 0(mod\ 5)$ Після припущення, отримаємо наступні перетворення, скористаючись Maple:

$$(0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2 + 3 \cdot i_3 + 4 \cdot i_4) - 10w,$$
$$0(i_0 - w) + 1(i_1 - w) + 2(i_2 - w) + 3(i_3 - w) + 4(i_4 - w),$$
$$x(v - w) + c(v - w) + m(w - w) + b(w - w) + n(s - w),$$
$$c(x + c) \cdot (v - w) + n(s - w),$$
$$(x + c)(v - w) + 3n(v - w),$$
$$(v - w)((x + c) + 3n).$$

Виконуючи подальші перетворення ми отримаємо порівняння [33] $(s - w) \equiv -2(v - w) \equiv 3(v - w)(mod\ 5)$. Обчислимо вираз $((x + c) + 3n)$ та продемонструємо отримані результати у таблиці 3.12.

Таблиця 3.12 – Номери індексів при $((x+c)+3n)$

n	x	c	$((x+c)+3n)$	n	x	c	$((x+c)+3n)$	n	x	c	$((x+c)+3n)$
0	1	2	3	1	2	4	4	3	0	4	3
0	1	3	4	1	3	4	0	3	1	2	2
0	1	4	0	2	0	1	2	3	1	4	4
0	2	3	0	2	0	3	4	3	2	4	0
0	2	4	1	2	0	4	0	4	0	1	3
0	3	4	2	2	1	3	0	4	0	2	4
1	0	2	0	2	1	4	1	4	0	3	0
1	0	3	1	2	3	4	3	4	1	2	0
1	0	4	2	3	0	1	0	4	1	3	1
1	2	3	3	3	0	2	1	4	2	3	2

Отримавши 30 доданків, що відповідають різним степеням елементу $\mu^{(v-w)}$, з яких степеню нуль – 10 (за кількістю нулів у таблиці 3.12), а першого степеню, другого, третього та четвертого – по 5. Звідси можна записати наступну рівність [33]:

$$a_v^2 a_w^2 a_s \cdot [10 \cdot \left(\mu^{(v-w)}\right)^0 + 5 \cdot \left(\mu^{(v-w)}\right)^1 +$$
$$+5 \cdot \left(\mu^{(v-w)}\right)^2 + 5 \cdot \left(\mu^{(v-w)}\right)^3 + 5 \cdot \left(\mu^{(v-w)}\right)^4] = a_v^2 a_w^2 a_s \cdot 1.$$

Так як будь-яка степінь $z = \mu^{(v-w)}$ відповідає рівнянню $z^4 + z^3 + z^2 + z = 1$, а $10\left(\mu^{(v-w)}\right)^0 = 0$, то вираз у квадратних дужках дорівнює одиниці. Це означає, що доданок $a_v^2 a_w^2 a_s$ буде входити до суми коефіцієнту b_r за умовою що $v \neq s \neq w(mod\ 5)$[33].

Якщо $|\{i_0, i_1, i_2, i_3, i_4\}| = 4$, то відмітивши різні чотири числа через v, w, s, k, ми отримаємо єдиний одночлен $a_v^2 a_w^2 a_s a_k$, для якого будемо мати $2v + w + s + k = 5r$, віднявши з цієї рівності $5v$, ми отримаємо

$(2v-2v)+(w-v)+(s-v)+(k-v)\equiv 0(mod\ 5)$, звідки $(w-v)+(s-$ $-v)+(k-v)\equiv 0(mod\ 5)$ та $(k-v)=-(w-v)-(s-v)$ [33].

Відмітимо номери індексів наступним чином: $i_y=w, i_c=s, i_e=p$, так що [33]:

$$(0\cdot i_0+1\cdot i_1+2\cdot i_2+3\cdot i_3+4\cdot i_4)-10v,$$
$$0(i_0-v)+1(i_1-v)+2(i_2-v)+3(i_3-v)+4(i_4-v),$$
$$y(w-v)+c(s-v)+e(p-v),$$
$$(y-e)(w-v)+(c-e)(s-v)(mod\ 5).$$

Відмітимо $w-v\equiv\gamma(mod\ 5)$, $s-v\equiv\delta(mod\ 5)$, і тоді $p-v\equiv(\gamma+\delta)\equiv\theta(mod\ 5)$[33]. Якщо не всі залишки γ,δ,θ відмінні, то при $\gamma=\delta=\theta$ ми будемо мати порівняність всіх цих трьох індексів v,w,s, і такий доданок не входить у суму, а при $\gamma=\delta\neq\theta$ отримаємо степінь елементу μ, яка дорівнює $(y+c-2e)\gamma$. Створюючи транспозицію y-го та c-гоелементу, отримаємо доданок з таким самим степенем елемента μ, так що ці доданки взаємно знищуються. При $w\equiv s, v\equiv p, p\equiv s$ доданку $a_v^2a_w^2a_sa_k$ не буде у сумі коефіцієнту b_r. Якщо всі залишки γ,δ,θ, різні та не дорівнюють нулю, то ми маємо всі ненульові залишки $\{\gamma,\delta,\theta,\rho\}=\{1,2,3,4\}$.

Сума всіх чотирьох різних ненульових залишків [33]:

$$\gamma+\delta+\theta+\rho=1+2+3+4=10\equiv 0(mod\ 5).$$

Звідси $\gamma+\delta+\theta=-\rho(mod\ 5)$, так як $\gamma+\delta+\theta\equiv 0(mod\ 5)$[33], то $-\rho\equiv 0(mod\ 5)$ та $\rho\equiv 0(mod\ 5)$, що є протиріччям, тому що ρ – це ненульовий залишок. Серед залишків γ,δ,θ є один нульовий, відмітимо його через $\theta=0$. Будемо мати $\gamma+\delta\equiv 0(mod\ 5)$, звідки:

$$0\cdot i_0+1\cdot i_1+2\cdot i_2+3\cdot i_3+4\cdot i_4=y(w-v)+0=y\cdot\gamma+c\cdot\delta(mod\ 5),$$

так як $\theta = p - v = 0,\ e(p - v) = 0$. Звідки ми маємо або $\delta \equiv 0(mod\ 5)$, або $\delta \neq 0(mod\ 5)$[33].

Дослідимо варіант, коли $\delta \equiv 0(mod\ 5)$. Якщо $\delta \equiv 0(mod\ 5)$, тоді

$$0 \cdot i_0 + 1 \cdot i_1 + 2 \cdot i_2 + 3 \cdot i_3 + 4 \cdot i_4 \equiv 0(mod\ 5).$$

Отже, $\mu^0 = 1$. Розподілити числа v, w, s, p за індексами i_0, i_1, i_2, i_3, i_4 можливо $5 \cdot 4 \cdot 3 = 60$ способами. Отже, якщо $w \equiv s \equiv v \equiv w(mod\ 5)$ та $2v + w + s + p = 5r$, то одночлен $a_v^2 a_w^2 a_s a_p$ не входить до суми для коефіцієнту b_r.

Дослідимо варіант, коли $\delta \neq 0(mod\ 5)$. Тоді $\gamma \equiv \delta \neq 0(mod\ 5)$, і при кожних $y \neq c$ будемо мати три доданки, тому отримаємо $\frac{60}{3} = 20$ доданків для подібних членів виду [33]:

$$a_v^2 a_w^2 a_s a_p \sum_{y \neq c} \mu^{y \cdot \gamma + c \cdot \delta} = a_v^2 a_w^2 a_s a_p \sum_{y=0}^{4} \mu^{y \cdot \gamma} \sum_{c \in \{y+1, y+2, y+3, y+4\}} \mu^{c \cdot \delta}.$$

Оскільки $\sum_{y=0}^{4} \left(\mu^{\delta}\right)^c = 0, \mu^0 = 1$, то $\sum_{y \neq c(mod\ 5)} \left(\mu^{\delta}\right)^c = \mu^{\delta \cdot y}$, так що отримаємо [33]:

$$a_v^2 a_w^2 a_s a_p \sum_{y \neq c} \mu^{y \cdot \gamma + c \cdot \delta} = a_v^2 a_w^2 a_s a_p \sum_{y=0}^{4} \mu^{y \cdot (\gamma + \delta)} = 5 \cdot a_v^2 a_w^2 a_s a_p.$$

В силу непарності числа 5 ми приходимо до висновків, що якщо $2v + w + s + p = 5r$, причому $w \neq v(mod\ 5)$, або $s \neq v(mod\ 5)$, або $p \neq v(mod\ 5)$, то одночлен $a_v^2 a_w^2 a_s a_p$ буде входити до суми коефіцієнту b_r [33].

Розглянемо випадок $|\{i_0, i_1, i_2, i_3, i_4\}| = 5$. Припустимо що $\{i_0, i_1, i_2, i_3, i_4\} = \{q, v, w, s, p\}$, тоді будемо мати $q + v + w + s + p = 5r \equiv$

$\equiv 0(mod\ 5)$, і ці різні числа можна розподілити за індексами i_0, i_1, i_2, i_3, i_4 відповідно всім $5! = 120$ підстановкам на п’яти елементах. Отримаємо набір $\{\bar{q}, \bar{v}, \bar{w}, \bar{s}, \bar{p}\} = \{\overline{i_0}, \overline{i_1}, \overline{i_2}, \overline{i_3}, \overline{i_4}\}$[33].

Розглянемо приклад, коли $\{\bar{q}, \bar{v}, \bar{w}, \bar{s}, \bar{p}\}$ менше п’яти, тобто $v \equiv w(mod\ 5), \bar{v} = \bar{w}$. Так як набори $\{q, v, w, s, p\}$ та $\{v, q, w, s, p\}$ є різними, з однаковим степенем для μ, ми можемо перейти до суми парної кількості доданків. Це – нуль над полем з двох елементів. Таким чином, при наявності яких-небудь порівнянь між індексами доданок $a_q a_v a_w a_s a_p$ не входить до складу суми коефіцієнту b_r [33].

Якщо потужність дорівнює п’яти, тобто всі числа q, v, w, s, p не порівняні за модулем п’ять, $\{q, v, w, s, p\} \equiv \{0,1,2,3,4\}(mod\ 5)$, то всі підстановки індексів розпадаються на комплекти з п’яти елементів $q \to q + 1(mod\ 5), v \to v + 1(mod\ 5)$ і т.д., де степінь μ є інваріантною. В таблиці 3.5 наведено всі степені $(A = 0j + 1v + 2w + 3s + 4k)$ такі, що $i_0 = 4$ (таблиця 3.13).

Таблиця 3.13 – Номери індексів степеня A

j	v	w	s	k			j	v	w	s	k		
0	1	2	3	4	A	$A(mod\ 5)$	0	1	2	3	4	A	$A(mod\ 5)$
4	0	1	2	3	20	0	4	1	2	0	3	17	2
4	0	1	3	2	19	4	4	1	3	0	2	15	0
4	0	2	1	3	19	4	4	2	1	0	3	16	1
4	0	2	3	1	17	2	4	2	3	0	1	12	2
4	0	3	1	2	17	2	4	3	1	0	2	13	3
4	0	3	2	1	16	1	4	3	2	0	1	11	1
4	1	0	2	3	19	4	4	1	2	3	0	14	4
4	1	0	3	2	18	3	4	1	3	2	0	13	3
4	2	0	1	3	17	2	4	2	1	3	0	13	3

Продовження таблиці 3.13

j	v	w	s	k			j	v	w	s	k		
4	2	0	3	1	15	0	4	2	3	1	0	11	1
4	3	0	1	2	14	4	4	3	1	2	0	11	1
4	3	0	2	1	13	3	4	3	2	1	0	10	0

З даних, які ми отримали у таблиці 3.13, ми можемо отримати наступне: $4\mu^0 + 5\mu^1 + 5\mu^2 + 5\mu^3 + 5\mu^4 = \mu + \mu^2 + \mu^3 + \mu^4 = 1$ у рамках поля $GF(2)$.

Розглянемо при яких випадках ми будемо мати одночлен виду $a_j a_v a_w a_s a_k$, який буде належати сумі коефіцієнта b_r [33]:

а) $q = v = w = s = k = r$, тоді утворюється одночлен виду a_r^5;

б) $w \neq v, w \equiv v(mod\ 5), 4v + w = 5r$, і тоді одночлен має вид $a_v^4 a_w$;

в) $w \neq s \neq v \neq w, w \neq s \neq v \neq w\ (mod\ 5)$, тоді утворюється одночлен виду $a_v^4 a_w^2 a_s$ або $a_v^3 a_w a_s$;

г) $w \neq v(mod\ 5), s \neq v(mod\ 5), k \neq v(mod\ 5)$, за умовою що $2v + w + s + k = 5r, v \neq s \neq k \neq w(mod\ 5)$,тоді ми отримаємо одночлен виду $a_v^2 a_w a_s a_k$;

д) одночлен виду $a_j a_v a_w a_s a_k$, за умовою що $j + v + w + s + k = 5r$.

Розглянемо для прикладу перші три доданки многочлену [33]:

$$u_0 = a_0^5,$$

$$u_1 = a_0^4 a_5 + a_0^3 a_1 a_4 + a_0^3 a_2 a_3 + a_1^3 a_2 a_0 + a_0^2 a_2^2 a_3 + a_0^2 a_1^2 a_3 + a_1^5,$$

$$u_2 = a_0^4 a_{10} + a_9 a_1 a_0^3 + a_8 a_2 a_0^3 + a_8 a_1^2 a_0^2 + a_7 a_3 a_0^3 + a_7 a_1^3 a_0 +$$

$$+ a_6 a_4 a_0^3 + a_2^2 a_0^2 a_6 + a_6 a_2 a_1^2 a_0 + a_6 a_1^4 + a_5 a_4 a_1 a_0^2 + a_5 a_3 a_2 a_0^2 +$$

$$+ a_5 a_2 a_1^3 + a_4 a_3^2 a_0^2 + a_4 a_3 a_2 a_1 a_0 + a_4 a_3 a_1^3 + a_4^2 a_2 a_0^2 + a_4^2 a_0 a_1^2 +$$

$$+ a_4 a_0 a_2^3 + a_4 a_1^2 a_2^2 + a_0 a_2^2 a_3^2 + a_2 a_3^2 a_1^2 + a_3^3 a_1 a_0 + a_3 a_1 a_2^3 + a_2^5 .$$

Зауважимо що у останній сумі пропущенні доданки: $a_5^2 a_0^3,\ a_7 a_2 a_1 a_0^2,\ a_6 a_1 a_3 a_0^2,\ a_5 a_3 a_0 a_1^2,\ a_5 a_0 a_1 a_2^2$ [33].

На прикладі переходів $x \to x^3, x \to x^5$, можна розглянути новий перехід $x \to x^\tau$, який буде здійснюватися у декілька етапів за допомогою розкладу τ на множники 3 та 5. Наприклад, перехід $x \to x^{15}$, згідно попереднім перетворенням, можна застосувати двома кроками: зробити перетворення $x \to x^3$,а далі зробити перетворення $x \to x^5$.

При дослідженні даного методу ми зробили висновок, що даний метод є базовою основою для створення алгоритму побудови незведеного многочлену $f_\tau(z)$ з залежністю коренів $z = x^\tau$, який може забезпечити підвищення криптостійкості даного алгоритму [34].

Під час дослідження ми зіштовхнулися з проблемою побудови незведених многочленів, що може бути матеріалом подальших досліджень. Тобто, спираючись на даний матеріал та розглянуту технологію можна окреслити перспективи подальших досліджень. Ми можемо дослідити будову загального алгоритму пошуку незведених многочленів з загальною степеневою залежністю коренів $z = x^\tau$ при τ – просте число [34].

3.3 Висновки за розділом 3

Отже, у розділі 3 ми провели дослідження сучасних алгоритмів шифрування, що використовують апарат модулярної арифметики та навели приклади програмної реалізації цих алгоритмів.

Ми дослідили шифр гамування. Виявили що з точки зору криптоаналізу, такий алгоритм шифрування є криптостійким. Тобто для абсолютної стійкості шифру гами необхідно мати повну випадковість ключа, рівність довжин ключа і відкритого тексту, також ключ має бути одноразовим. Наеви приклади роботи шифру гами. Провели дослідження роботи скремблера. Навели приклади програмної реалізації роботи скремблера в Java та в Maple.

Також ми дослідили ще один алгоритм шифрування, що здійснюється за допомогою генерації незведених многочленів. Ми дослідили особливість даного методу на базі переходів $x \to x^3$ та $x \to x^5$. Такий метод генерації є базовою основою для створення алгоритму побудови незведеного многочлену $f_\tau(z)$ з залежністю коренів $z = x^\tau$, який може забезпечити підвищення криптостійкості даного алгоритму шифрування.

ВИСНОВКИ

Отже при дослідженні алгоритмів криптографії та криптоаналізу, що використовують апарат модулярної арифметики у алгоритмах криптографії, у першому розділі, ми ознайомилися з основними поняттями криптографії та модулярної арифметики.

У другому розділі ми дослідили історію виникнення криптографії, класичні шифри, основні шифри та принцип дії таких шифрів. Навели приклади шифрування інформації за допомогою шифрів Цезаря та Вернама, принцип дії яких будується на основі теорії чисел, а саме модулярної арифметики. Також, ми провели аналіз сучасного стану досліджень у криптографії та криптоаналізу, отримавши необхідні результати та висновки для подальших досліджень.

У третьому розділі ми провели програмну реалізацію деяких алгоритмів шифрування. А саме, ми розглянули шифр гамування, роботу скремблера та генерацію незведених многочленів. Ми навели приклади програмної реалізації роботи скремблера в Java та Maple. Також ми дослідили ще один алгоритм шифрування, що здійснюється за допомогою генерації незведених многочленів. Ми дослідили особливість даного методу на базі переходів $x \to x^3$ та $x \to x^5$. Такий метод генерації є базовою основою для створення алгоритму побудови незведеного многочлену $f_\tau(z)$ з залежністю коренів $z = x^\tau$, який може забезпечити підвищення криптостійкості даного алгоритму шифрування.

Отже на сьогоднішній день, коли інформація охоплює інтернет простір, її захист є важливим регламентом якісної безпеки. Алгоритми шифрування, які використовують модулярну арифметику мають стійкість, тобто можуть протистояти спробам криптоаналітика зламати криптосистему.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ященко В. В., Нестернко Ю. В. Введение в криптографию : учебник, 4-е издание. Москва : МЦНМО, 2012. 348 с.

2. Вербіцький О. В. Вступ до криптології : підручник. Львів : видавництво науково-тех. літ., 1998. 247 с.

3. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття : підручник. Львів : БаК, 2003. 144 с.

4. Фільштінський В. А., Бережний А. В. Математичні основи криптографії . Суми : Сумський державний університет, 2011. 138 с.

5. Погорелова Б. А. Словарь криптографических терминов. Москва : МЦНМО, 2006. 91 с.

6. Алфёров А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. StudFiles : файловый архив для студентов. URL : https://studfiles.net/preview/6311470/ (дата звернення 28. 04. 2020).

7. Оглобліна О. І., Сушко Т. С., Шрамко Ю. В. Елементи теорії чисел : навч. посібник. Суми : Сумський державний університет, 2015. 186 с.

8. Нестерова Л. Ю., Карпенкова Н. В. Создание криптографии с помощью модулярной математики. *Молодой ученый*. 2014. № 21.1. С. 237–240.

9. Богуш В. М., Мухачов В. А. Криптографічні застосування елементарної теорії чисел : навч. посібник. Київ : ДУІКТ, 2006. 126 с.

10. Бабак В. П. Теоретичні основи захисту інформації : підручник. Київ : НАУ, 2008. 752 с.

11. Бабенко Л. К., Ищукова Е. А. Принципиальные особенности проведения дифференциального криптоанализа блочных шифров. Известия ЮФУ. Технические науки. 2009. №11. URL: https://cyberleninka.ru/article/n/printsipialnye-osobennosti-provedeniya-differentsialnogo-kriptoanaliza-blochnyh-shifrov (дата звернення 29. 04. 2020).

12. Горбенко И. Д., Лавриненко Д. И. Эффективная реализация операции возведения в степень больших целых чисел в криптографических системах методом предвычислений по фиксированному основанию. *Радиоэлектроника и информатика*. 1999. №4 (9). URL: https://cyberleninka.ru/article/n/effektivnaya-realizatsiya-operatsii-vozvedeniya-v-stepen-bolshih-tselyh-chisel-v-kriptograficheskih-sistemah-metodom-predvychisleniy (дата звернення 12. 05. 2020).

13. Мельничук Е. М., Новоселов С. А. Характеристические многочлены некоторых гиперэллиптических кривых родов 2,3 и p-ранга 1. *ПДМ. Приложение*. 2019. №12. URL : https://cyberleninka.ru/article/n/harakteristicheskie-mnogochleny-nekotoryh-giperellipticheskih-krivyh-rodov-2-3-i-r-ranga-1 (дата звернення 17. 05. 2020).

14. Романьков В. А. Новая семантически стойкая система шифрования с открытым ключом на базе RSA. *ПДМ*. 2015. №3 (29). URL : https://cyberleninka.ru/article/n/novaya-semanticheski-stoykaya-sistema-shifrovaniya-s-otkrytym-klyuchom-na-baze-rsa (дата звернення 22. 05. 2020).

15. Умаров Ш. А., Акбаров Д. Е. Разработка нового алгоритма шифрования данных с симметричным ключом. *Журнал СФУ. Техника и технологии*. 2016. №2. URL : https://cyberleninka.ru/article/n/razrabotka-novogo-algoritma-shifrovaniya-dannyh-s-simmetrichnym-klyuchom (дата звернення 30. 05. 2020).

16. Бабенко Л. К., Голотин Д. В. Об основных особенностях функционирования и реализации поточного шифра Trivium. *Известия ЮФУ. Технические науки.* 2015. №5 (166). URL : https://cyberleninka.ru/article/n/ob-osnovnyh-osobennostyah-funktsionirovaniya-i-realizatsii-potochnogo-shifra-trivium (дата звернення 25. 06. 2020).

17. Бабенко Л. К., Сидоров И. Д. Параллельный алгоритм дискретного логарифмирования методом решета числового поля. *Известия ЮФУ. Технические науки.* 2008. №8. URL : https://cyberleninka.ru/article/n/parallelnyy-

algoritm-diskretnogo-logarifmirovaniya-metodom-resheta-chislovogo-polya (дата звернення 27. 06. 2020).

18. Галов К.А., Черкесова Л.В., Сафарьян О.А. Реализация алгоритма Миллера-Рабина на языке C#. *Молодой исследователь Дона.* 2019. №1 (16). URL : https://cyberleninka.ru/article/n/realizatsiya-algoritma-millera-rabina-na-yazyke-c (дата звернення 28. 06. 2020).

19. Востоков С. В., Востокова Р. П., Беззатеев С. В. Теория чисел и приложения в криптографии. *Чебышевский сборник*. 2018. №3 (67). URL : https://cyberleninka.ru/article/n/teoriya-chisel-i-prilozheniya-v-kriptografii (дата звернення 30. 06. 2020).

20. Бабенко Л. К., Ищукова Е. А., Маро Е. А., Сидоров И. Д., Кравченко П. П. Развитие криптографических методов и средств защиты информации. *Известия ЮФУ. Технические науки*. 2012. №4. URL : https://cyberleninka.ru/article/n/razvitie-kriptograficheskih-metodov-i-sredstv-zaschity-informatsii (дата звернення 12. 07. 2020).

21. Рыжков А. В. Протокол стойкого шифрования по разделяемому ключу малого размера в группе точек эллиптической кривой. *Интеллектуальные технологии на транспорте.* 2016. №3. URL : https://cyberleninka.ru/article/n/protokol-stoykogo-shifrovaniya-po-razdelyaemomu-klyuchu-malogo-razmera-v-gruppe-tochek-ellipticheskoy-krivoy (дата звернення 17. 07. 2020).

22. Агеенко А. Н. Аналог системы шифрования RSA на платформе m2(Zn). *Вестник ОмГУ.* 2009. №4. URL : https://cyberleninka.ru/article/n/analog-sistemy-shifrovaniya-rsa-na-platforme-m2-zn (дата звернення 29. 07. 2020).

23. Виноградова Т. А. Алгоритм генерации гиперэллиптических кривых на основе комплексного умножения. *Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки.* 2007. №10. URL : https://cyberleninka.ru/article/n/algoritm-generatsii-giperellipticheskih-krivyh-na-osnove-kompleksnogo-umnozheniya (дата звернення 30. 07. 2020).

24. Романьков В. А. Диофантова криптография на бесконечных группах. *ПДМ*. 2012. №2 (16). URL : https://cyberleninka.ru/article/n/diofantova-kriptografiya-na-beskonechnyh-gruppah (дата звернення 27. 08. 2020).

25. Скобелев В. В. «Ленточная» теорема и ее приложения. *ПДМ.* 2009. №4 (6). URL : https://cyberleninka.ru/article/n/lentochnaya-teorema-i-ee-prilozheniya (дата звернення 30. 08. 2020).

26. Павлюк Д. Н., Тимошкин А. И., Тимошкин А. А., Трошков М. А., Шовгарян А. Г., Токарев И. С., Ткаченко А. С. Математические основы шифрования информации несимметричным алгоритмом на основе эллиптических кривых в конечном поле простых чисел. *Известия вузов. Северо-Кавказский регион. Серия: Технические науки.* 2007. №3. URL : https://cyberleninka.ru/article/n/matematicheskie-osnovy-shifrovaniya-informatsii-nesimmetrichnym-algoritmom-na-osnove-ellipticheskih-krivyh-v-konechnom-pole-prostyh (дата звернення 07. 09. 2020).

27. Самойленко Д. В., Финько О. А. Помехоустойчивая криптосистема, основанная на Китайской теореме об остатках, для n каналов с шумом и имитирующим злоумышленником. *Известия ЮФУ. Технические науки.* 2010. №11. URL : https://cyberleninka.ru/article/n/pomehoustoychivaya-kriptosistema-osnovannaya-na-kitayskoy-teoreme-ob-ostatkah-dlya-n-kanalov-s-shumom-i-imitiruyuschim-zloumyshlennikom (дата звернення 08. 09. 2020).

28. Кистанов А. М. Разработка криптосистемы на основе эллиптической кривой над конечным иррациональным полем. *Вестник Самарского государственного технического университета. Серия: Технические науки*. 2005. №33. URL : https://cyberleninka.ru/article/n/razrabotka-kriptosistemy-na-osnove-ellipticheskoy-krivoy-nad-konechnym-irratsionalnym-polem (дата звернення 17. 09. 2020).

29. Бабенко Л. К., Ищукова Е. А. Дифференциальный криптоанализ упрощенной функции хэширования sha. *Известия ЮФУ. Технические науки.* 2010. №11. URL : https://cyberleninka.ru/article/n/differentsialnyy-kriptoanaliz-uproschennoy-funktsii-heshirovaniya-sha (дата звернення 19. 09. 2020).

30. Остапов С. Е., Євсеєв С. П., Король О. Г. Технології захисту інформації: навч. посібник. Харків: ХНЕУ, 2013. 476 с.

31. Франчук В. М. Захист інформаційних ресурсів : криптографічні та стеганографічні методи захисту даних: посібник для викладачів, вчителів та студентів інформатичних спеціальностей. Київ : НПУ ім. М. П. Драгоманова. Інститут інформатики, 2012. 120 с.

32. Задірака В. К., Олексик О. Комп'ютерна криптологія : підручник. Київ, 2002. 505 с.

33. Титов С. С., Торгашова А. В. Генерация неприводимых многочленов, связанных степенной зависимостью корней. *Доклады ТУСУР*. 2010. №2-1 (22). URL: https://cyberleninka.ru/article/n/generatsiya-neprivodimyh-mnogochlenov-svyazannyh-stepennoy-zavisimostyu-korney (дата звернення 25. 09. 2020).

34. Сушко С. О., Кузнецов Г. В., Фомичова Л. Я., Корабльов А. В. Математичні основи криптоаналізу : навч. посібник. Дніпропетровськ : НГУ, 2004. 391 с.

ДОДАТОК А

Програмна реалізація роботи скремблера в Java

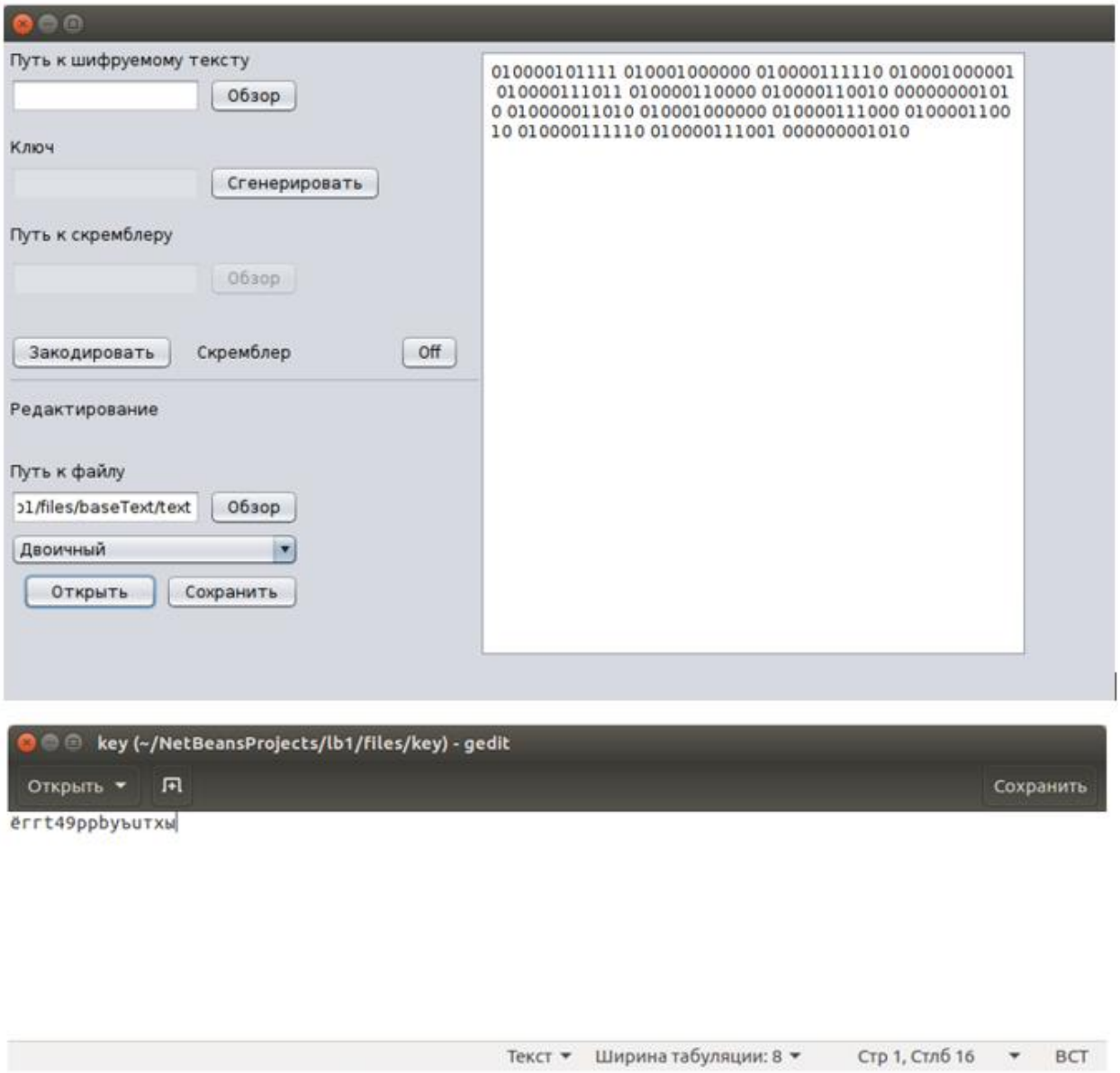

Рисунок А.1 – Вихідний текст у двійковому виді, представлений в Java

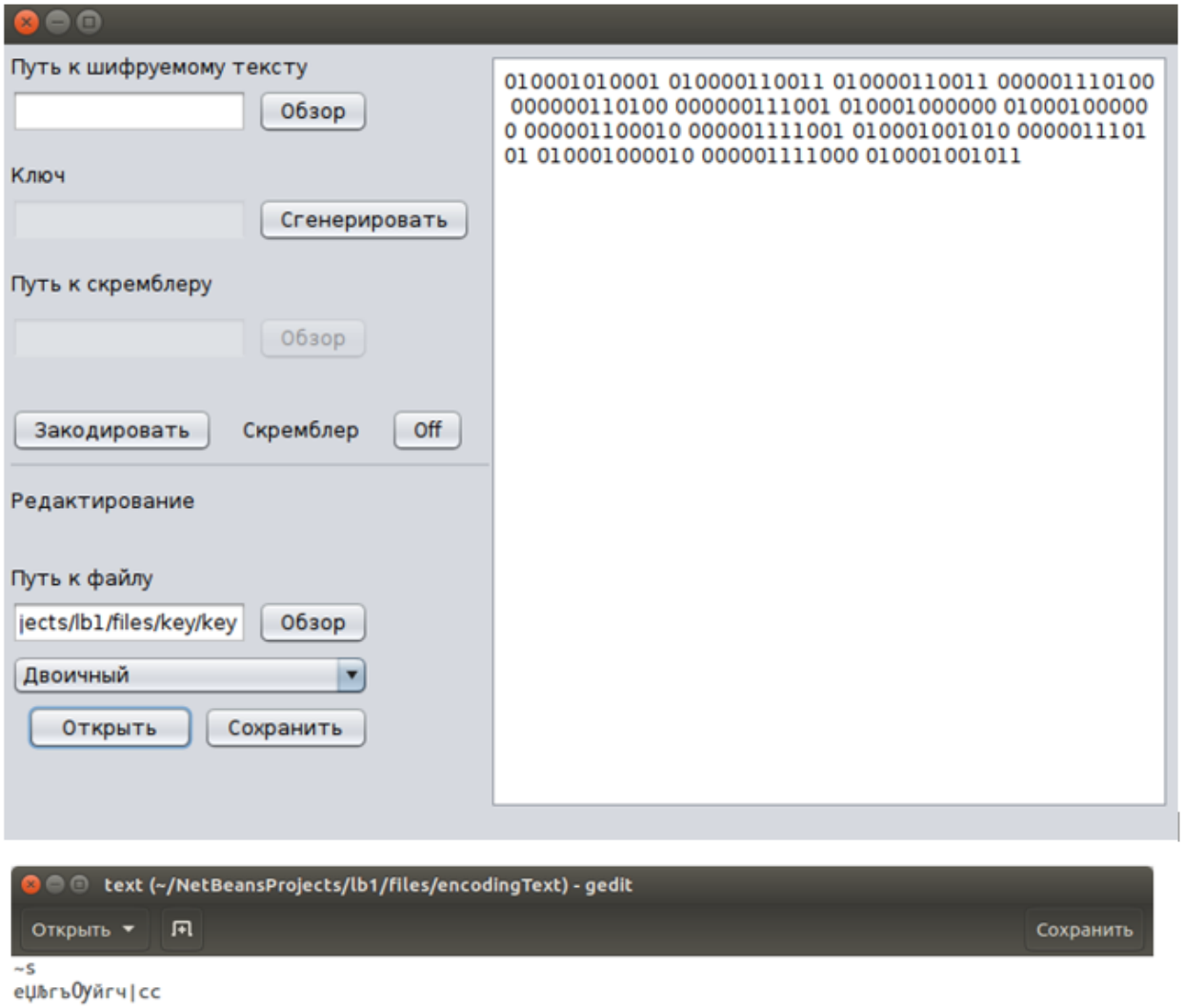

Рисунок А.2 – Ключ у двійковому виді, представлений в Java

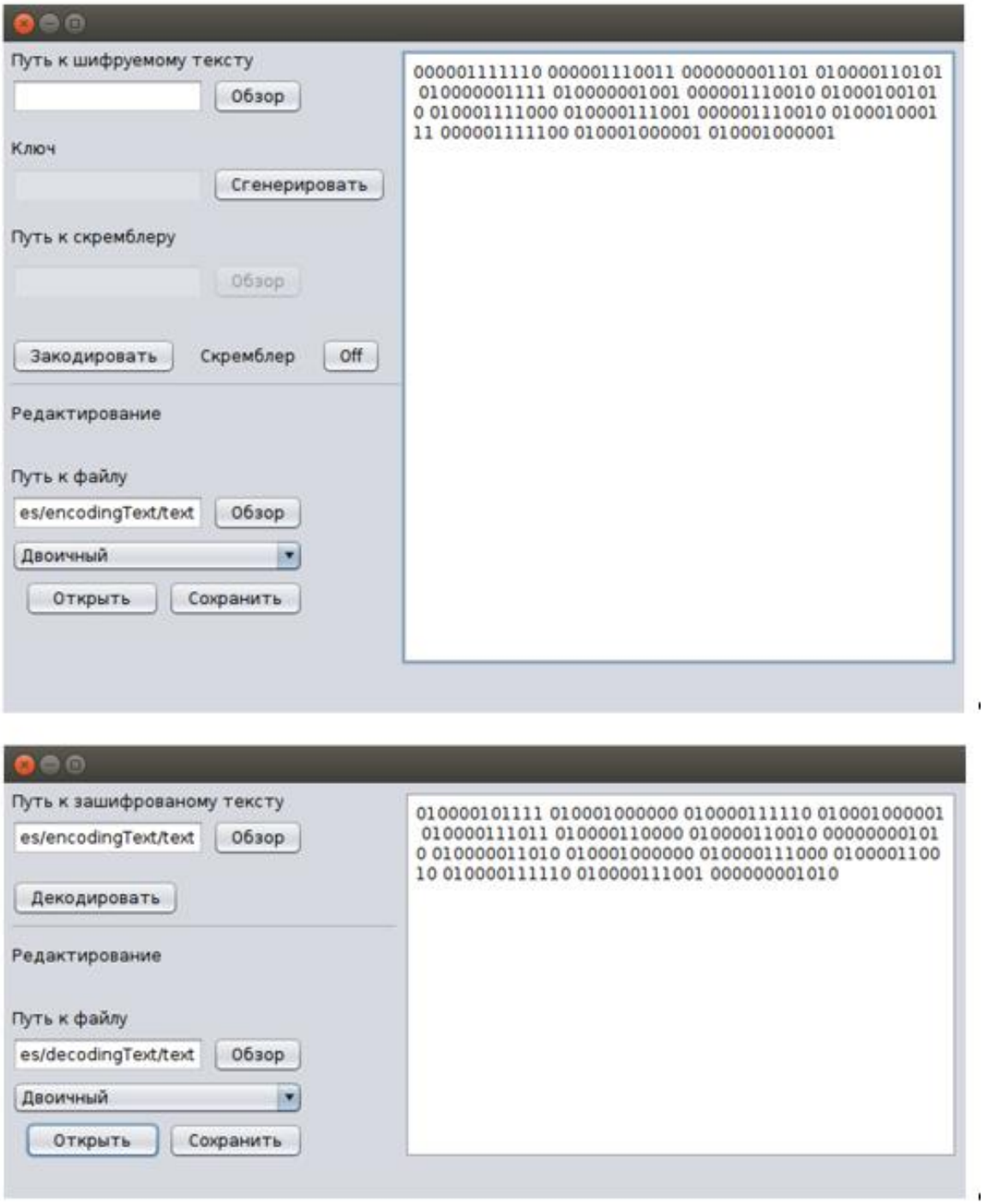

Рисунок А.3 – Шифрограма у двійковому виді в Java

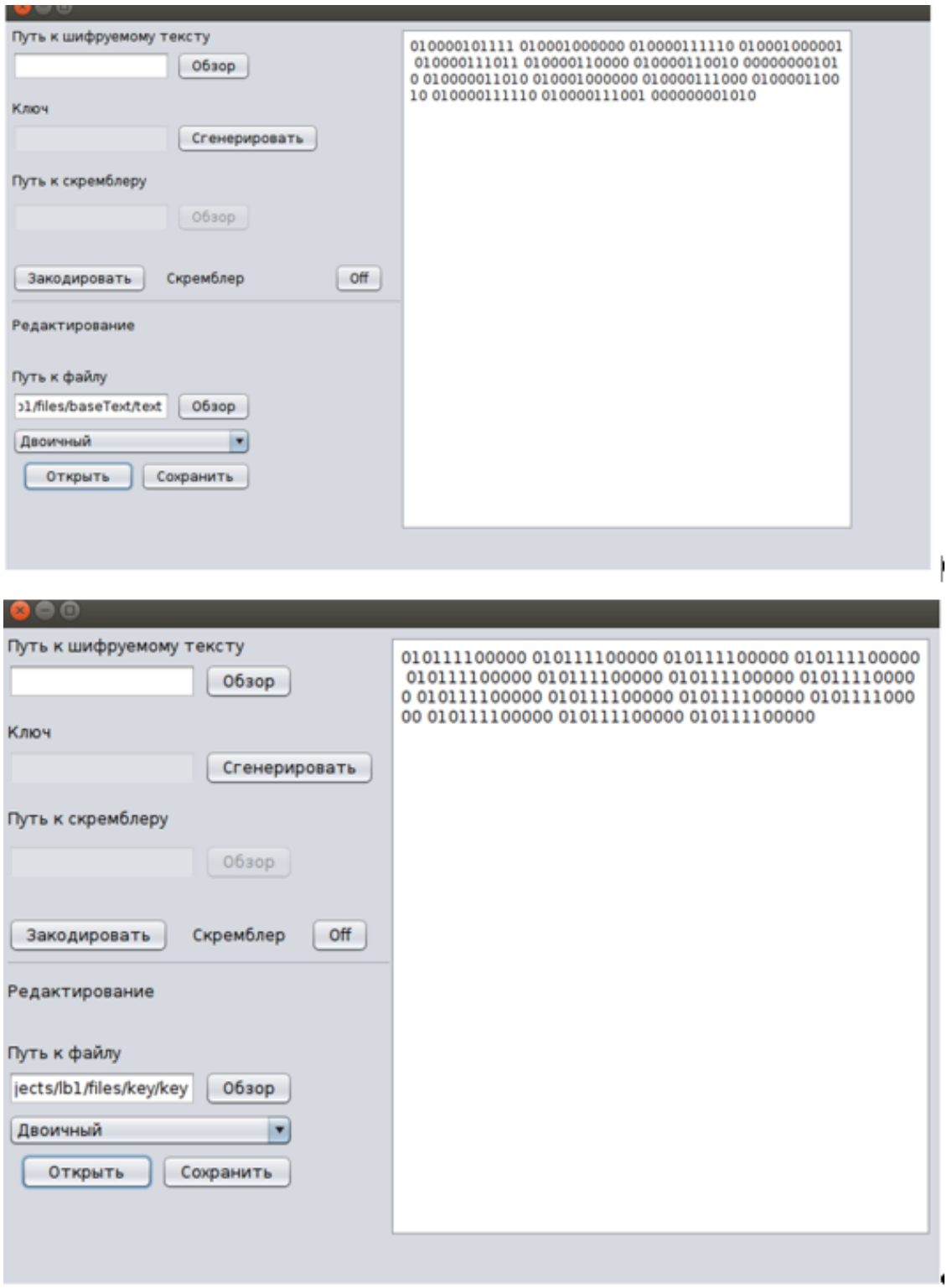

Рисунок А.4 – Співставлення двійкового коду в Java

ДОДАТОК Б

Програмна реалізація роботи скремблера в Maple

```
> restart;
> x := array(1..32): y := array(1..32): z := array(1..32):
>
>
  x[1] := 1: x[2] := 2: x[3] := 3: x[4] := 4: x[5] := 5: x[6] := 6: x[7] := 7: x[8] := 8:
  x[9] := 9: x[10] := 10: x[11] := 11: x[12] := 12: x[13] := 13: x[14] := 14: x[15] := 15: x[16] := 16:
  x[17] := 17: x[18] := 18: x[19] := 19: x[20] := 20: x[21] := 21: x[22] := 22: x[23] := 23: x[24] := 24:
  x[25] := 25: x[26] := 26: x[27] := 27: x[28] := 28: x[29] := 29: x[30] := 30: x[31] := 31: x[32] := 32:


>
  y[1] := "А": y[2] := "Б": y[3] := "В": y[4] := "Г": y[5] := "Д": y[6] := "Е": y[7] := "Ё": y[8] := "Ж":
  y[9] := "З": y[10] := "И": y[11] := "Й": y[12] := "К": y[13] := "Л": y[14] := "М": y[15] := "Н": y[16] := "О":
  y[17] := "П": y[18] := "Р": y[19] := "С": y[20] := "Т": y[21] := "У": y[22] := "Ф": y[23] := "Х": y[24] := "Ц":
  y[25] := "Ч": y[26] := "Ш": y[27] := "Щ": y[28] := "Ы": y[29] := "Ь": y[30] := "Э": y[31] := "Ю": y[32] := "Я":
```

```
> for i from 1 to 32 do
    z[i] := convert(x[i], binary):
  end do:
  print(`x= `,x): print(` y=`,y): print(` z=`,z):
                    x= [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32]
y=
  ["А", "Б", "В", "Г", "Д", "Е", "Ё", "Ж", "З", "И", "Й", "К", "Л", "М", "Н", "О", "П", "Р", "С", "Т", "У", "Ф", "Х", "Ц", "Ч", "Ш", "Щ", "Ы", "Ь", "Э", "Ю",
  "Я"]
z= [1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011, 10100, 10101, 10110, 10111, 11000, 11001, 11010, 11011,
  11100, 11101, 11110, 11111, 100000]
```

```
> s := "ПРИВЕТЯМЕРЗЛИКИНАВАЛЕРИЯ";
  CODE := "":
  v := array(1..length(s)/2):

                                        s := "ПРИВЕТЯМЕРЗЛИКИНАВАЛЕРИЯ"
```

```
> for i from 1 to length(s)/2 do
      v[i] := s[2·(i − 1) + 1..2·i]:
      for j from 1 to 32 do
        if (v[i] = y[j]) then
                          #print(z[j]);
                           CODE := cat(CODE, z[j], " ");
        end if:
      end do:
  end do:
  print(CODE);
```

```
  print(CODE);
"10001 10010 1010 11 110 10100 100000 1110 110 10010 1001 1101 1010 1100 1010 1111 1 11 1 1101 110 10010 1010 100000 10001 10010 1010 11 110 10100
  100000 1110 110 10010 1001 1101 1010 1100 1010 1111 1 11 1 1101 110 10010 1010 100000 "
```

Рисунок Б. 1 – Програмна реалізація роботи скремблера в Maple